National Bureau
of Standards

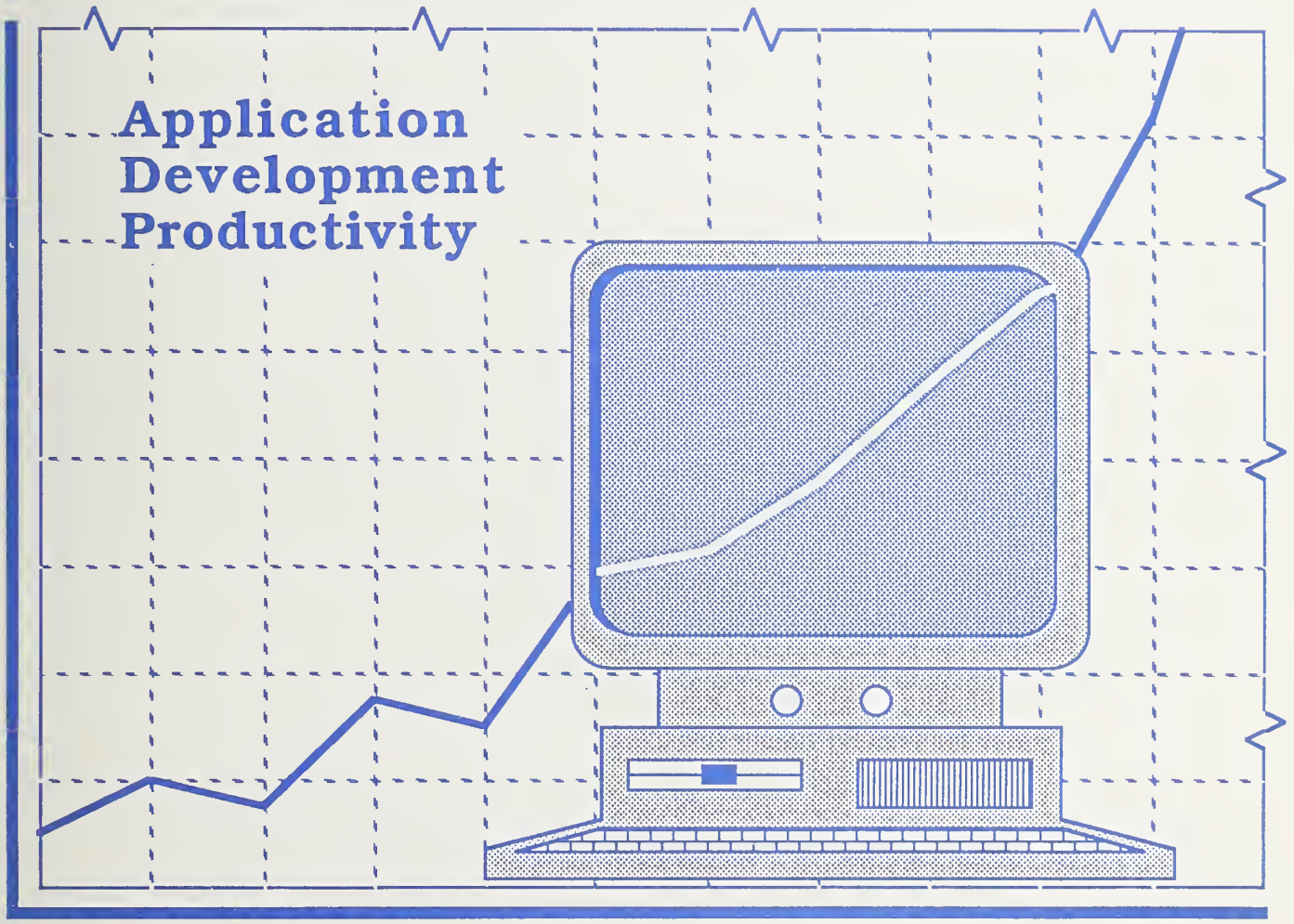
NBS
PUBLICATIONS

Computer Science and Technology

NBS Special Publication 500-143

Guide to the Selection and Use of Fourth Generation Languages

Martha Mulford Gray



The National Bureau of Standards[1] was established by an act of Congress on March 3, 1901. The Bureau's overall goal is to strengthen and advance the nation's science and technology and facilitate their effective application for public benefit. To this end, the Bureau conducts research and provides: (1) a basis for the nation's physical measurement system, (2) scientific and technological services for industry and government, (3) a technical basis for equity in trade, and (4) technical services to promote public safety. The Bureau's technical work is performed by the National Measurement Laboratory, the National Engineering Laboratory, the Institute for Computer Sciences and Technology, and the Institute for Materials Science and Engineering.

The National Measurement Laboratory

Provides the national system of physical and chemical measurement; coordinates the system with measurement systems of other nations and furnishes essential services leading to accurate and uniform physical and chemical measurement throughout the Nation's scientific community, industry, and commerce; provides advisory and research services to other Government agencies; conducts physical and chemical research; develops, produces, and distributes Standard Reference Materials; and provides calibration services. The Laboratory consists of the following centers:

- Basic Standards[2]
- Radiation Research
- Chemical Physics
- Analytical Chemistry

The National Engineering Laboratory

Provides technology and technical services to the public and private sectors to address national needs and to solve national problems; conducts research in engineering and applied science in support of these efforts; builds and maintains competence in the necessary disciplines required to carry out this research and technical service; develops engineering data and measurement capabilities; provides engineering measurement traceability services; develops test methods and proposes engineering standards and code changes; develops and proposes new engineering practices; and develops and improves mechanisms to transfer results of its research to the ultimate user. The Laboratory consists of the following centers:

- Applied Mathematics
- Electronics and Electrical Engineering[2]
- Manufacturing Engineering
- Building Technology
- Fire Research
- Chemical Engineering[2]

The Institute for Computer Sciences and Technology

Conducts research and provides scientific and technical services to aid Federal agencies in the selection, acquisition, application, and use of computer technology to improve effectiveness and economy in Government operations in accordance with Public Law 89-306 (40 U.S.C. 759), relevant Executive Orders, and other directives; carries out this mission by managing the Federal Information Processing Standards Program, developing Federal ADP standards guidelines, and managing Federal participation in ADP voluntary standardization activities; provides scientific and technological advisory services and assistance to Federal agencies; and provides the technical foundation for computer-related policies of the Federal Government. The Institute consists of the following centers:

- Programming Science and Technology
- Computer Systems Engineering

The Institute for Materials Science and Engineering

Conducts research and provides measurements, data, standards, reference materials, quantitative understanding and other technical information fundamental to the processing, structure, properties and performance of materials; addresses the scientific basis for new advanced materials technologies; plans research around cross-country scientific themes such as nondestructive evaluation and phase diagram development; oversees Bureau-wide technical programs in nuclear reactor radiation research and nondestructive evaluation; and broadly disseminates generic technical information resulting from its programs. The Institute consists of the following Divisions:

- Ceramics
- Fracture and Deformation [3]
- Polymers
- Metallurgy
- Reactor Radiation

[1]Headquarters and Laboratories at Gaithersburg, MD, unless otherwise noted; mailing address Gaithersburg, MD 20899.
[2]Some divisions within the center are located at Boulder, CO 80303.
[3]Located at Boulder, CO, with some elements at Gaithersburg, MD.

Computer Science and Technology

NBS Special Publication 500-143

Guide to the Selection and Use of Fourth Generation Languages

Martha Mulford Gray

Center for Programming Science and Technology
Institute for Computer Sciences and Technology
National Bureau of Standards,
Gaithersburg, MD 20899

September 1986

U.S. DEPARTMENT OF COMMERCE
Malcolm Baldrige, Secretary

National Bureau of Standards
Ernest Ambler, Director

Reports on Computer Science and Technology

The National Bureau of Standards has a special responsibility within the Federal Government for computer science and technology activities. The programs of the NBS Institute for Computer Sciences and Technology are designed to provide ADP standards, guidelines, and technical advisory services to improve the effectiveness of computer utilization in the Federal sector, and to perform appropriate research and development efforts as foundation for such activities and programs. This publication series will report these NBS efforts to the Federal computer community as well as to interested specialists in the academic and private sectors. Those wishing to receive notices of publications in this series should complete and return the form at the end of this publication.

Library of Congress Catalog Card Number: 86-600582
National Bureau of Standards Special Publication 500-143
Natl. Bur. Stand. (U.S.), Spec. Publ. 500-143, 65 pages (Sept. 1986)
CODEN: XNBSAV

U.S. Government Printing Office
Washington: 1986

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, DC 20402

TABLE OF CONTENTS

LIST OF FIGURES

PREFACE

This report has been prepared by the Institute for Computer Sciences and Technology (ICST) in response to requests for information and guidance from Federal Government and private sector organizations over the past year. ICST formulated a schedule of tasks designed to organize and research the area of Fourth Generation Languages (4GLs). This report is the third publication issued in this area. The first was a summary of the **Application Development Productivity Workshops**, held on November 13-15, 1985 at the National Bureau of Standards. The first day of the workshop was a plenary session on 4GLs. Much of the discussion centered on an urgent requirement for information and guidance on 4GLs. This impetus set the stage for accelerating the dissemination of information and eventually led to the publication of the second publication from this research area.

The second report was **A Functional Model for Fourth Generation Languages**, NBS Special Publication 500-138, by Gary E. Fisher [FISH86a], issued in June 1986. "The purpose of this functional model is to define Fourth Generation Language in a manner similar to specifying the functions of a specific software application. This definition process allows managers, technical personnel, and end-users to refer to a commonly understood terminology in the 4GL context. In addition, the interfaces between 4GLs and external entities (i.e. humans, operating systems, peripheral devices, and other application systems) can be identified and studied for research purposes and possible standardization" [FISH86].

An additional report, **A Guideline for Choosing an Applications Development Approach**, by James Hall, is being developed. This guideline describes the various options that have evolved for the development of computer applications, including the impact of 4GLs on development methodologies.

This report represents not only the efforts of the author but also a tremendous amount of input from Gary E. Fisher and James Hall. The author acknowledges their guidance and support.

EXECUTIVE SUMMARY

This report provides guidance on the selection process for Fourth Generation Languages (4GLs); a description of the features, functions, and capabilities of 4GLs; and a brief discussion of the use of 4GLs. It is aimed at managers who are in the process of selecting an appropriate 4GL, determining appropriate applications for the 4GL environment, and planning to avoid potential problems of 4GL misuse.

After a comparison of the selection issues for third and Fourth Generation languages and a summary of the 4GL functional model, the ten step selection process listed below is recommended.

1. Describe the problem or application
2. Analyze the application environment
3. Decide on the selection approach
4. Define the application requirements
5. Develop a list of desired 4GL features
6. Rate the desired features
7. Select candidate packages
8. Screen and rate the candidate 4GLs selecting the top few candidates
9. Analyze the top candidates in detail
10. Select a 4GL

Sections detailing the analysis of the application environment include; the hardware environment, the software environment, the organizational environment, and the user environment. A checklist to assist managers in this analysis of the application environment, step two of the selection process, is also given.

A thorough discussion of the product selection criteria follows. There are separate sections on operating features, user interface features, language features, security features, report writing features, data management features, graphics features, and implementation issues. Checklists of general questions for the screening process and lists of suggested questions for more detailed analyses are included in these sections.

The final section provides recommendations on 4GL use, including a discussion of how 4GLs are currently used, types of applications in which they are used, and factors which aid successful 4GL implementation.

1.0 INTRODUCTION

Currently there are hundreds of software products in the marketplace that are called Fourth Generation Languages (4GLs). Selecting one of these products for use in any environment involves a myriad of factors. Since the definition of 4GLs is still under debate, what a 4GL should do is also clouded. The literature often describes 4GLs as saviors to data processing management and solutions to programming backlogs, invaluable tools for professional programmers and a great gift to end-users. Unfortunately, 4GLs have not proved to be the panacea for all data processing ills.

There are four major areas of concern related to 4GL usage:

- Performance - hardware resource consumption, response times, multi-user access, real-time processing
- Portability - no 4GL standards, portability of language, portability of developed code, number of skilled programmers, number of revisions and upgrades
- Support - training, maintenance, transferability of skills, vendor support
- Relevance - applicability to organization, problem or application, and methodology for software development

However, when an apropros 4GL is selected for a suitable 4GL application and used in an atmosphere of sound data processing management, 4GLs can provide a useful tool for application development.

This report attempts to provide guidance on the selection process for 4GLs: a description of the features, functions, and capabilities of 4GLs; and a brief discussion on the use of 4GLs, including potential pitfalls of using 4GLs improperly.

1.1 SELECTION ISSUES OF THIRD AND FOURTH GENERATION LANGUAGES

The literature is filled with definitions and debates over what is or isn't a 4GL. Descriptions have been written about four generations of computer languages (See **A Functional Model for Fourth Generation Languages** [FISH86]), usually based on the progression of computer hardware and machine, assembler and compiler languages. There has been little discussion of the language attributes that differentiate compiler or third generation languages (3GLs) from 4GLs other than the procedural and non-procedural aspects. Analyzing the selection issues relating to 3GLs and 4GLs highlights some other important differences in these two generations of languages.

David Spencer [SPEN85] contends that -- "To belong to the fourth-generation, a language must have crossed the threshhold of a world where the programmer specifies the task to be done to the point where the knowledge of how to do the task is contained in the language itself." This means that the language of 4GLs contains certain functionality that is not present in 3GL. The difference is immediately clear when one looks at the selection issues of 3GL and 4GL.

John Cugini in **Selection and Use of General-Purpose Programming Languages - Overview** [CUGI84a] presented a survey of selection factors for Ada*, BASIC, C, COBOL, FORTRAN, Pascal, and PL/1. He stated that the selection criteria were based on three major factors, "1)the language and its implementation, 2)the application to be programmed, and 3)the user's existing facilities and software," and presented criteria for these three factors. His Table of Contents listed the features of the languages that are described in the report. Some of the language factors are:

- Syntactic Style
  - Statement Terminator
  - Fixed or Free Format
  - Statement Labels
  - Identifiers
  - Implicit or Declared Entities
  - Program Length
- Semantic Structure
  - Control of Execution
    - Structured Programming
    - Blocks
    - Subroutines
    - Functions
    - Recursion
    - Generic Procedures
    - Exception Handling
    - Concurrency
  - Control of Data
    - Storage Classes
    - External Data
    - Data Abstraction
  - Packages
- Data Types and Manipulation
  - Checking and Coercion
  - Elementary Data
    - Numeric
    - Character
    - Logical
    - Bit
    - Pointer
  - Aggregate Data
    - Arrays
    - Files and I/O
    - Records
    - Sets

*Ada is a registered trademark of the U.S. Government, Ada Joint Project Office.

These issues listed for 3GL selection are not the primary issues for 4GL selection. No longer does a manager look at how the computer is accomplishing the job nor even care if the computer is capable of accessing bit data. The questions facing data processing managers in assessing 4GLs are removed from this level of detail because the languages are a generation above this level. Only in specialized cases where systems are being optimized are these issues being addressed. The 4GL handles many of the issues listed in the table of contents without most users' knowledge.

Perhaps a better example of the additional functionality of a 4GL is a simple example of 4GL code versus 3GL code. In this example hypothetical 4GL statements are translated into an equivalent BASIC program.

Hypothetical 4GL

```
USE CUSTOMER-INFO
FIND ZIP-CODE › 02134 AND ‹ 02500
PRINT CUSTOMER-NAME ADDRESS CITY STATE-CODE ZIP-CODE
```

BASIC

```
100 when exception in
110   open #1: name  "CUSTOMER_INFO", access input, rectype display
120 use
130   print "'CUSTOMER_INFO' file cannot be opened."
140   exit handler
150 end when
160 when exception in
170    do
180      input #1, if missing then exit do:   &
&           CUSTOMER_NAME$, ADDRESS$, CITY$, STATE_CODE$, ZIP_CODE
200      if ZIP_CODE › 02134 AND ZIP_CODE ‹ 02500 THEN
210         print CUSTOMER_NAME$, ADDRESS$, CITY$, STATE_CODE$, ZIP_CODE
220      endif
230    loop
240 use
250   print "Cannot perform print process."
260 end when
270 close #1
280 end
```

These two examples would accomplish the same goal of printing customer names and addresses with specified zip-codes but the amount of code necessary in BASIC is significantly greater than in a 4GL. In addition, it is easier to conceptually understand the three lines of 4GL code than the 19 lines of BASIC code. A

4GL generally handles error messages, opening and closing files (for the USE command), the formating of text to the screen (for the PRINT command), and a certain amount of IF-THEN logic (for the FIND command). The user no longer has to tell the computer how to accomplish the given task but simply tells the computer what is wanted. Thus, a number of selection issues that are important for 3GL are simply not considerations for 4GL.

1.2 4GL FUNCTIONAL MODEL

A Functional Model for Fourth Generation Languages [FISH86] defines the functions of a 4GL and presents a "commonly understood terminology" for the discussion of 4GLs. According to this report:

> *"the capabilities provided by 4GLs are grouped into three major areas based on similarities in overall function. They are--*
>
> - *User functions;*
> - *Data management functions; and*
> - *System functions.*
>
> *User functions define those capabilities necessary to provide a high level dialogue between the 4GL and users of the 4GL. Users of 4GLs may include humans and other systems. ... This area is further broken down into the following specific functions:*
>
> - *Screen formatting*
> - *Menu management*
> - *Message prompting*
> - *Logical device management for devices such as light pen, touch-screen, mouse, graphics tablet, remote sensors, etc.*
>
> *Data management functions provide capabilities to describe, store and retrieve, and perform ancillary tasks in the management and safekeeping of application data. ... Functions in this area include the following:*
>
> - *Logical data structure management*
> - *Data storage and retrieval*
> - *Archiving and restoration*
> - *Auditing*
> - *Data security*

System functions provide the support services necessary to allow the user of 4GLs to define and access applications in relation to the constraints of the environment in which the 4GL operates. ... These functions include, but are not limited to, file handling, job control, communications, and other applications."

In addition to these three areas of functions which define a minimal 4GL, [FISH86] also describes other capabilites which may be added to the 4GL to make it an advanced 4GL such as:

- o graphics capabilities,
- o a programming language interface,
- o a command language for direct access to the operating system,
- o program/data/text editing capabilities,
- o a debugger/compiler,
- o real-time control language functions and services,
- o office automation facilities, and
- o word processing capabilities.

This summary of **A Functional Model for Fourth Generation Languages** [FISH86] should help define the basic functions of a 4GL. The application requirements that are going to be addressed by the 4GL determine whether minimal or advanced functions are necessary.

The presence or absence of the functions listed above needs to be determined for the selection of a 4GL. The generalization of functions in [FISH86] necessitates a view of 4GLs that is completely independent of any implementation or component aspect of actual 4GLs. However, the feature analysis portion of the selection process requires translating these generalized functional properties into specific features that may be found among 4GLs existing in today's marketplace. This results in a perspective that will be more appropriate for the selection of a 4GL.

2.0 SELECTION PROCESS

According to most of the current literature, the selection process for 4GLs is a fairly simple matter: define the problem or application (or develop a needs analysis), analyze the users, evaluate the commercial packages, and purchase the right package. The authors simply suggest that the selector choose the package which matches the hardware required and appears to match the requirements of the application. There has been very little written on how all of this is accomplished, however.

Because a 4GL is not a simple package generally obtained to solve only one problem but is software which incorporates user functions, data management functions, and system functions, the importance of the selection process should not be underestimated. Of course, the level of effort involved in the selection process will vary greatly depending on the size of the application. It would not be cost effective to spend hundreds of staff hours completing the selection process for a micro-based 4GL costing less than $1000. By the same logic, it would not be cost effective to make a hasty selection decision for a software package costing hundreds of thousands of dollars and affecting the corporate database.

The main purpose of this part of the report is to identify the ten steps in the 4GL selection process and to provide some guidance on the selection process as a whole. This section will not present details on every aspect of the process but will try to put the process in perspective. For example, this report will not detail how to establish selection and evaluation teams but will reference the literature for this area (see [WILS86]). This section will not cover the procurement process because this varies from agency to agency. The report will however, give guidance that should lead users through the selection process to the point where the procurement process can begin.

2.1 OVERVIEW

The selection process for a 4GL (summarized in figure 1) is a ten step process that should be completed for large or small procurements. The level of effort spent on each step will vary but all of these steps should be completed before procuring any 4GL. The selection process will be described briefly followed by examples of the levels of effort involved for different kinds of applications.

The selection process begins with a brief description of the problems that are to be solved or the applications that are to be handled by the 4GL. This first step should include a brief description of the scope and domain of the problem and the applications to be addressed.

10 STEP 4GL SELECTION PROCESS

STEP	MAJOR CONCEPTS
1. **DESCRIBE PROBLEM OR APPLICATION**	o scope o purpose o domain
2. **ANALYZE APPLICATION ENVIRONMENT**	o hardware environment o software environment o organizational environment o user environment
3. **DECIDE ON SELECTION APPROACH**	o establish selection and evaluation team o identify decision makers and control points
4. **DEFINE REQUIREMENTS**	o identify application requirements
5. **DEVELOP LIST OF DESIRED 4GL FEATURES**	o see Product Selection Criteria (Section 3.0)
6. **RATE DESIRED FEATURES**	o establish users rating system o identify mandatory features o identify undesirable features
7. **SELECT CANDIDATE PACKAGES**	o survey literature o survey DP software reference services o attend conferences, trade shows
8. **RATE 4GLs AND SELECT TOP FEW**	o screen using hardware, operating systems, & mandatory features o use other ratings to narrow selection
9. **ANALYZE TOP FEW IN DETAIL**	o benchmark o pilot test o demonstrations o user comments
10. **SELECT 4GL**	

Figure 1

The **second step** is to complete an analysis of the application environments, i.e. the hardware, software, organizational, and user environments. A more complete description of these environments follows this overview and a checklist for completing this step is provided in Appendix A.

The **third step** is to decide how the selection decision will be made, i.e. if there will be a selection and evaluation team or a single decision-maker.

The **fourth step** is to define the requirements for the 4GL based on the information gathered in the first two steps. It is important here that what are defined are actually requirements and not unrealistic dreams, wishes, desires, etc.

After the requirements have been defined, a list of 4GL features which can fulfill these requirements must be developed. This is **step five** of the selection process. Section five of this report covers the product selection criteria that should be considered in this step.

Step six is to develop a ranking or rating system for the 4GL features so that the most desired features can be differentiated from the least desired. Mandatory features should be identified in this step plus any features which would make a package unacceptable.

Step seven consists of selecting candidate packages. Information on the availability of 4GLs can be obtained from the computer literature, software reference services, trade shows, computer conferences, and computer user groups.

Step eight is to eliminate most of the candidate packages. First, a screening process should be established to eliminate packages that do not fulfill the hardware requirements, operating system requirements, or mandatory feature requirements, or possess undesirable features. A suggested screening form is included in Appendix B that could be used for this process. After the screening process, further elimination should be based on rating the other features of the 4GLs and comparing these ratings with the user ratings of the desired features. This step should eliminate all but a few packages.

Step nine is to analyze the top few candidate packages in detail. Approaches such as benchmarking, pilot testing, vendor demonstrations, and gathering user opinions can be used for this step.

The final step, **step ten**, is to select the most appropriate package.

The level of effort involved in these steps varies with the size of the application, the size of the hardware and software, etc. An example of **Step one** for a small, micro-based, application is:

> Develop a new program to determine what employees in the division will be eligible for step increases in the next fiscal year. The domain of the problem is budget or personnel software and the scope is the division.

An example of a larger problem would be to:

> Develop a new program to determine the impact of new, complex retirement legislation on agency-wide staffing levels. The domain of the problem is personnel software and the scope is agency-wide.

Since a large 4GL implementation is not usually considered for solving just one problem or handling one application, larger procurements might involve repeating Step one for numerous problems or applications.

Step two, the analysis of the hardware, software, organizational, and user environments, often changes the perspective of the problem. The result is a more realistic, practical description of the requirements for the 4GL. Less effort should be expended on this step for a small procurement with a limited number of users than on a large, multi-user implementation. This step nevertheless needs to be completed for any selection.

The importance of these steps cannot be overstated. Thomas Nies in [NIES86] credits St. Thomas of Aquinas as saying, "A little error in the beginning becomes a great error in the end." This is especially relevant for these steps of the selection process. If these analyses are not completed, a 4GL may be purchased which will not solve the problem, will not fit into the organization, or will not serve the users needs. William Gruber [GRUB85] states "There is no such thing as the 'best' software: there are only suitable and unsuitable packages for a given application." If the environments of the 4GL are not analyzed, a suitable package cannot be found.

The **third step** in the selection process is deciding how the selection decision for the 4GL will be made. For small, micro-based implementations it may be appropriate to have one individual identified to make this decision. If so, that person should be identified and the approval process established. For large applications many organizations may want to establish selection and evaluation teams to handle the rest of the process. During this step the individuals who will be on the selection and evaluation team should be identified. The team is usually composed of users, data processors and management representatives

[see WILS86]. At this point, some organizations may want to look at the 'make or buy' decision and begin work on a cost justification. If the process is going to involve a formal procurement a committee or individual should be assigned here.

Steps four and five, defining the requirements and developing a list of desired 4GL features, are less time consuming for small applications than for large applications. The amount of effort certainly multiplies with the number of problems or applications involved. The important factor in these steps, however, is the identification of true requirements, (not dreams, wishes, etc.) and the identification of specific 4GL features that will fulfill the requirements of the applications. Small implementations may find the features listed on the screening form (see Appendix A) sufficient for detailing the desired 4GL features. Selection and evaluation teams for large implementations may need to thoroughly review Section 3 of this report and may want to consider using a number of the additional questions listed to help identify needed 4GL features.

The next step, **Step six**, is to develop a ranking or rating system for these features. For large applications, the selection and evaluation team may want to have users rate the features of the 4GLs. These ratings have to be combined to develop the list of desired features with a single rating for each feature. For small applications with one decision-maker, that decision-maker may want to survey the users or, if there is only one user, simply rate the features.

The selection and evaluation team for a large procurement may want to establish a weighing system for the features to indicate which features are more important than others. For any size application, mandatory features have to be identified. These are features that must be present for the 4GL to fulfill the requirements. Features which are definitely not wanted or undesirable must also be identified.

Step seven begins the process of selecting candidate packages. There are software reference services, articles in the computer literature, conferences, and trade shows which focus on 4GL products. Initially there may be hundreds of packages which appear to be candidates even for small applications. At this point all of them have to be screened to insure completeness.

Step eight is the screening of the packages and elimination of those that do not fit the desired features. A suggested screening form is given in Appendix B that could be used for this process. The screening form can be altered to match the desired features for the specific problem or application. The idea is to initially eliminate packages based on the hardware requirements, operating system requirements, mandatory features and/or

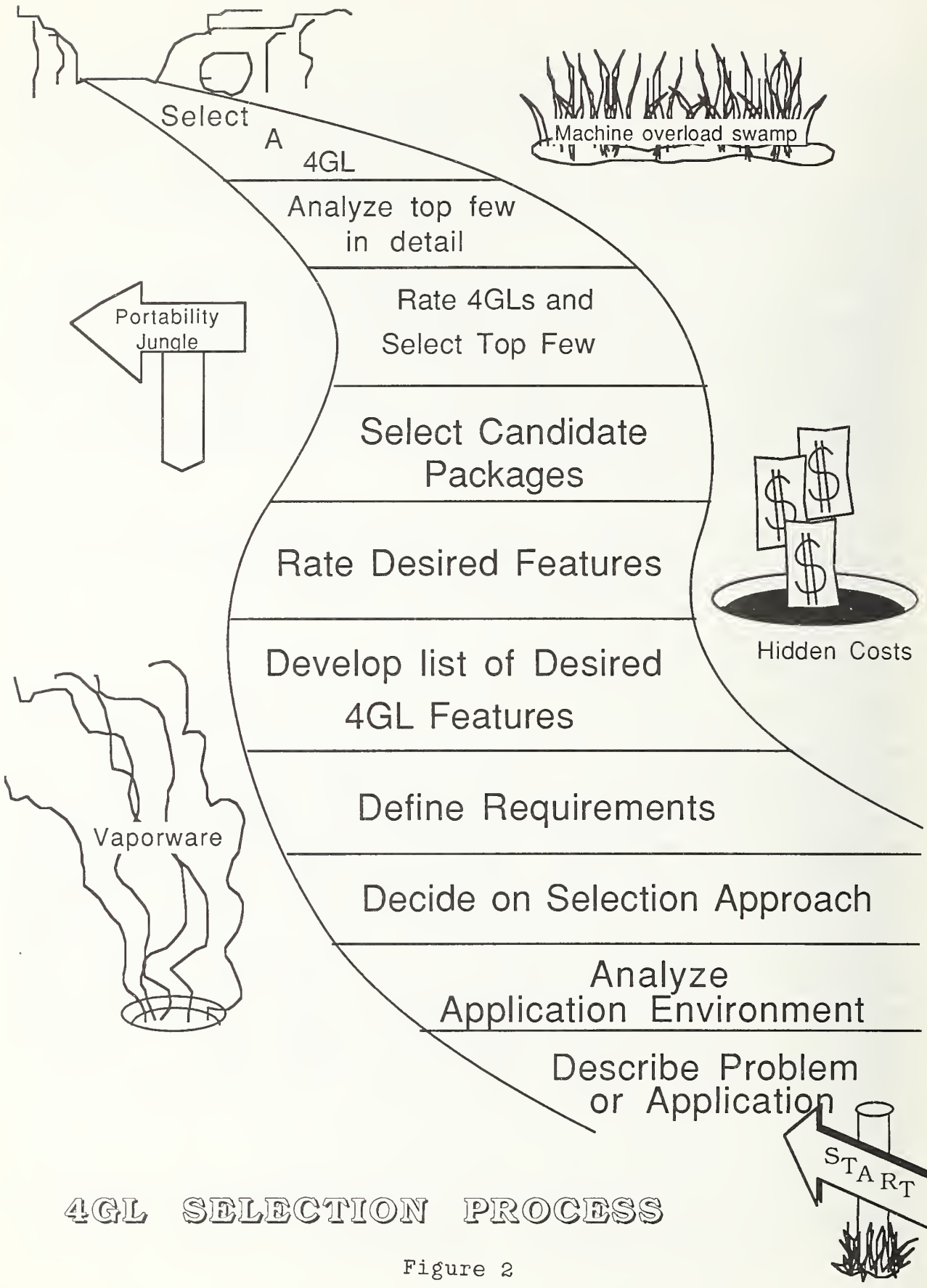
undesirable features. If the hardware and operating system requirements are known, the initial elimination can be made on these requirements. If they are not known, the elimination has to be made on the mandatory features and undesirable features. Then further elimination can be performed using the ratings of the other features. This step needs to eliminate all but a few packages.

Step nine is to analyze the top few candidate packages in detail. There are various approaches that can be used such as benchmarking, pilot testing, demonstrations, and gathering other user opinions. For large applications all of these approaches might be appropriate. The major consideration here is the relative cost of the packages under consideration. If this is a large procurement, a fair amount of time and effort should be spent on this step. Benchmarks and pilot tests may not be appropriate for small procurements.

All of these steps should lead to **step ten** which is the selection of a specific package. In the 4GL world, these ten steps are sometimes fraught with worries and concerns (see figure 2) but they at least provide a safe course to follow.

2.2 APPLICATION ENVIRONMENT

Most of the literature on 4GL selection states that a careful analysis of the problem to be solved or the applications that will use the 4GL or 4GLs should be undertaken. There are few attempts, however, to provide a structure for this analysis. If the whole selection process is viewed as solving a puzzle, then certain pieces of information need to be provided before specific products can be evaluated. Otherwise the solution will never be found. A data processing person such as the manager of the data processing facility, head of the information center or whoever traditionally reviews the requests for data processing should take a first look at the environment for the 4GL. Further information will be filled in as subsequent factors are evaluated, but the initial cut on this part should be attempted by someone from the data processing facility. Some information can only be obtained by analyzing the applications for which the 4GL is being purchased and analyzing the environment in which the 4GL will reside. Much of this information relates only to the existing environment.



4GL SELECTION PROCESS

Figure 2

2.2.1 Hardware Environment

Knowledge of the hardware environment is critical to the 4GL selection process. In some cases the hardware environment is already established and is not going to change dramatically during the acquisition of a 4GL. In this case a 4GL must be selected that will be compatible with the existing hardware. There is no point in purchasing 4GL features that will be unusable.

In other cases the hardware is part of the acquisition plan and will be adjusted for the acquisition of a 4GL. Some 4GLs have been hardware resource consumptive and caused machine overload problems. When hardware acquisition plans have not been adjusted for the 4GL, performance of the existing system has deteriorated. Thus the performance issues of 4GLs will need to be considered in relation to the hardware environment.

Appendix A is a checklist of information that should be obtained during the analysis of the application environment. The first information identifies current or planned hardware that will be utilizing the 4GL. This includes the processor, (mainframe, mini, or micro); the memory that is or will be available; the peripheral equipment such as printers, graphics devices, terminals, storage devices, and modems; the hardware and software links to other computers and networks; office automation equipment which might interface with the application; and hardware security requirements. The requirements of these devices need to be specified.

If the application or problem to be solved is a stand-alone, micro-based project, some of the information for the checklist will be fairly simple to determine. As the size of the project increases, the degree of difficulty for obtaining the information increases.

2.2.2 Software Environment

The most important issue relating to the software environment is the operating system. A number of 4GL products interface with the operating system to allow users to perform certain functions such as file handling and storage allocation. It is important to specify under what operating system the 4GL must operate. Again, future acquisition and upgrade plans must be taken into consideration.

The next issues of concern are language interfaces for the 4GL. If there is existing code in other languages, e.g. third or fourth generation languages, that will be utilized, the 4GL will have to be able to interface with these languages. For example,

if there is existing COBOL code that handles verification of users and data, this may not need to be rewritten but simply called from the 4GL. The 4GL would have to be able to interface with COBOL. At the same time, there may be other existing programs that will not be rewritten but will have to interface with the 4GL, at least in the beginning of 4GL utilization, to insure a smooth transition.

The next items on the checklist (Appendix A) are the database and data dictionary systems that are currently installed. Most 4GL products include a database system and a data dictionary or data definition facility. To insure a smooth transition and eliminate duplicating data and data entry effort, these existing database and data dictionary systems should interface with the 4GL. It is critical to specify the requirements of these existing systems. It is also important to identify any existing file handling routines since these may have to interface with the 4GL.

In addition to the software that manages the data, the data that exists either in automated form or manual form must be identified. Some of the information that needs to be gathered includes the size of the existing files, the size of the records, size of the tables, and types of data that are available. The current security requirements of these files and the programs that handle the files must be documented.

Analyzing the software environment serves one other purpose, i.e. insuring that there is no software, already installed, which could serve the current needs. Quite often one part of an organization does not know about the computer functions in another part of an organization. Thus, there could be software already installed which would serve the needs of the application or problem at hand.

2.2.3 Organizational Environment

The current structure of the organization has an impact on the policies and procedures of the data processing functions. Top management needs to support the implementation of new technology and should set the priorities and assist in defining the information needs [MART86]. However, what management should do is, quite often, not what is done. The structure of the organization and the style of management of the organization must be considered realistically. The purchase of a 4GL alone is not necessarily going to trigger major changes in the management of an organization.

An appropriate person needs to seriously analyze the structure of the organization, where decision points are located, where policies are established, where users may be located, and how

these will remain according to the long-range plan of the organization. This analysis serves two functions. First, it insures that the 4GL product selected will fit into current operating procedures. There are certain products that are designed for information center or end-user use that would be decidedly inappropriate for a highly structured, controlled organization with numerous decision points for each access to the corporate data. This is not just a matter of analyzing the users but also the way data processing is handled in the organization. Richard Wilson, in "The 4GL Evaluation Team" [WILS86], suggests the following:

> *"any evaluation of a product that affects the organization as a whole must be conducted by people with a thorough understanding of that organization, its corporate culture, the key players, and their relationships. Above all, the evaluation must be made in terms of the organization's basic business objectives. How a product affects the organization determines the evaluation approach and method."*

The second function that an organizational analysis serves is to help identify users and potential users for the 4GL. A selection team will need to be assembled for large projects. The team must include users and potential users of the 4GL. For a small, stand-alone system, the user should consider the source of the input data or recipient of the output data, organizationally. Quite often the result of this organizational analysis for stand-alone systems highlights that very few systems are really stand-alone. Usually the input data comes from another source and the output data goes to another person or system.

Ronald Shelby and Ronald Dubien [SHEL85] suggest defining "the type of data processing organizational structure in terms of centralized or decentralized management responsibility" and identifying "the extent of data sharing" before purchasing a DBMS. The same approach applies to 4GL. If you cannot analyze the whole management structure of the organization at least looking at these two areas will provide some useful information. Some products are most appropriate for numerous users, sharing information and databases while others are more appropriate for stand-alone situations or highly controlled environments.

2.2.4 Other Environmental Factors

There are a few other areas that should generally be described before beginning the analysis of the user environment. This involves taking a look at the overall data processing environment. The type of applications that usually run on the facility should be considered. Could they be called commercial, scientific, or a mix of both? This helps to get a view of how

the 4GL will fit with the rest of data processing for the organization. Are most applications highly time critical or fairly flexible? If most applications are run with severe time constraints, the efficiencies of the 4GLs must be weighed heavily so that the impact on the rest of the data processing is minimized. Are the applications mostly routine, repetitious applications such as weekly reports or are most applications ad-hoc, interactive requests? Are there heavy or high volume transactions? Who are the users of most of these systems?

The reason for studying the overall environment for the 4GL is to minimize the adverse impact the 4GL selection decision could have on the rest of the organization. The fit of the 4GL within the data processing environment must be considered.

2.2.5 User Environment

The literature generally describes dividing users into two groups, end-users or non data processing types, and data processing professionals. James Martin in [MART85a] and [MART85b] suggests grouping users into three groups, end-users, programmers and analysts. I suggest that none of these are sufficent for really analyzing the user environment. These groupings are not adequate for determining the kind of usage a 4GL will have.

A different method for approaching this user environment analysis is proposed based on the research described below. This method produces user categories which are more appropriate for a 4GL environment. A matrix for using these categories is given in Appendix A.

2.2.5.1 Query Language User Categories

Matthias Jarke and Yannis Vassiliou [JARK86] suggest a much more complex grouping of users in their research on "a systematic approach to matching categories of query language interfaces with the requirements of certain user classes..." Jarke uses "four dichotomous classifications: familiarity with programming concepts, frequency of query language usage, knowledge about the application, and range of operations required." Jarke considers the first two, familiarity with programming concepts and frequency of query language usage, "syntatic knowledge" or the user's ability to technically interact with the system. When he rated users on this knowledge, he derived three user types: novice, skilled and professional.

The second two classifications, knowledge about the application and range of operations ("how many different types of queries the user requires"), he considered "semantic knowledge" or the task

structure of the user. Jarke rated users using these criteria and derived four user types: 1)casual users, 2)clerical users, 3)managerical users and 4)application specialists. Casual users have only a general idea about the structure and content of the database and require a limited range of needed operations. Clerical users usually only perform a limited range of operations but may have in-depth knowledge of the database and application. Managerial users have a large range of operations they may want to perform and want to waste little time learning the database or system. Applications specialists have detailed knowledge of the database and application and require the ability to perform a large range of operations.

Jarke then created a matrix with the novice, skilled, and professional categories as rows and the casual, managerial, clerical and applications specialists as columns. In each block of the matrix he placed the features of a query language that seemed appropriate. For example what features a novice managerial user would need or a professional applications specialist. This creates twelve blocks of varying features.

2.2.5.2 4GL User Categories

Individuals trying to select a 4GL do not necessarily need to establish an elaborate rating system for users, but might want to utilize some of the ideas relating to this research. It is not sufficient for example, to only think of end-users or even of novice users. Are the users management types who will use the 4GL infrequently and thus have limited time or patience to spend learning the commands of the 4GL? Are there clerical types who will generate numerous reports using the 4GL and command runs that will be written by someone else? Are there casual users who, no matter how much they know about programming and how much they know about the application, will need help using the 4GL because they will only be using the 4GL every couple of months? By the same logic, there may be end-users who after using the 4GL daily become application specialists. It is appropriate to think of users in terms of knowledge of programming, knowledge of 4GLs, knowledge of the application, and rate or kind of usage.

This kind of analysis has a large impact on the kind of features of a 4GL that might be desired. Figure 3 illustrates a smaller matrix that shows some features of 4GL and how they fit into this scheme. The categories of casual and managerial are combined because there are many similarities in the 4GL features that are required. The term "clerical" has been changed to operational because this is a better fit for the kinds of 4GL users.

This matrix simply illustrates that different features of 4GL are appropriate for different kinds of users. Since the users need to be included on any selection and evaluation team, or the views

of the users included in the selection process, the selector needs to identify who the users are. The users need to weigh or rank the features of 4GL that are most important or the features that they expect to see included with the 4GL. If the users have not been clearly identified, this is not possible.

The other purpose of this analysis of the users is to provide a sense of reality to a list of desires presented by the users. Quite often there is a great difference between what users want and what they need. By analyzing the different types of users and the interactions of the users with the system, the selector can better determine what is really needed and make certain to include features that the users may have forgotten. For example, novice managerial type users would not know that a preprogrammed user profile would provide a way of leading the user through the 4GL to accomplish set tasks.

Application Knowledge & Functionality

Programming Skills & Knowledge of 4GL	Managerial Users	Operational Users	Application Specialists
Novice	Pre-programmed User Profiles Help facilities User leading menus Read only file protection	Pre-programmed User Profiles Menus Help facilities Predefined Reports Reod only file protection	Menus Help Facilities Novice mode Diagnostic messages
.	.	.	.
.	.	.	.
.	.	.	.
.			
.			
Professional	Non-procedurol user longuage Spreodsheet Downloading Dota to Micro	Non-procedurol user longuage User friendly report writer Word processing Office Auto. interfaces	Technical user language Code generator Debugger Compiler Communications Expert mode

User Assessment For 4GL Features
Figure 3

3.0 PRODUCT SELECTION CRITERIA

The following paragraphs describe features which should be considered in Step 5 of the selection process. A description of the features, importance of the features, and concerns relating to these features are included. Following each feature discussion are questions that should be asked during the screening process, Step 8 of the selection process, and additional questions that can be used when analyzing the top few candidates in detail, Step 9 of the selection process. Not all questions will be appropriate for all applications but these are suggestions that can be used where the application requirements dictate.

3.1 OPERATING FEATURES

This section describes some of the features that are basic to the operation of a 4GL. They include the hardware and software requirements for a 4GL, portability attributes, performance characteristics, communication features and operating system interfaces. Information on these features would be mandatory before any selection decision could be made, no matter what the application area. The checklist given at the end of this section should be completed for any software product being considered.

Basic information on every software product being considered must be known before any further judgements can be made. This information is needed for the very first step of the selection process --- weeding out potential packages based on hardware, operating system and mandatory feature requirements. The decision-maker needs to be careful in the approach to gathering the information, however. For example, it is not sufficient to ask whether or not a 4GL will run on existing hardware. The appropriate question should be what hardware versions of the 4GL are currently available or installed. The answer to the second question gives a decision-maker not only information on whether the package will run in the current hardware environment but also on the potential for portability if the hardware environment changes. The answer also indicates the scope of the user environment and perhaps the scope of the support available for the 4GL.

Some of the other operating features are those that compose the interface between the 4GL and the operating system and software environment. These include aspects of the operating system that influence the behavior of the 4GL. For example, the 4GL can provide self-explanatory error messages in place of operating system messages. The 4GL can warn the user if a query or other process will consume hardware resources or take a long time. The 4GL can restart automatically after a system crash and might even

be able to restart at the task level. The 4GL can identify data involved in a crash. The 4GL can handle system defaults and let the users override the defaults if desired. All of these features of a 4GL involve the 4GL interface with the operating system. The importance of these features is dependent on the hardware and software environment of the 4GL and the level of expertise of the users. A user who is not familiar with any operating system commands, diagnostics, etc., may panic if the system crashes unless the 4GL can handle some of these situations.

There are also many file handling features of 4GL that involve the 4GL interface with the operating system. These are described in [FISH86] as follows:

> *"File handling includes file management, transfer of files, file editing, and other actions that affect files as a whole. File management at the system level allows the user to set and retrieve pertinent information about files such as file access modes (i.e. read, write, extend), buffer size, file size, file location, directory entries, etc. Files may be transferred from one device to another such as in copying from fixed disk to a removable disk. Operating systems may include programs to edit files based on physical structure rather than logical structure as is most often found in 4GLs."*

Examples follow that utilize these functions. User procedures may need to be stored and cataloged. Users may want to save the results of a 4GL query for later processing such as specialized printing. Users may need to create new files, allocate additional storage space or re-use deleted space. Users may want to utilize the windowing capabilities of their operating environment to window files or procedures.

One other important area should be considered in this section, networking and communication capabilities. This report will not begin to address all of the issues in networking computers and communications. If the hardware and software environment includes requirements for networking and communications then these features need to be studied in detail to insure that the 4GL will allow for these requirements.

Below is a checklist for features relating to the operation of the 4GL. This should probably be the first information gathered on any candidate 4GL.

OPERATING FEATURES CHECKLIST

SCREENING INFORMATION

On what hardware does it execute? _____

_____

On what operating system does it execute? _____

_____

How much memory is required:

	MINIMUM	SUGGESTED
Main memory		
Hard Disk		
Floppy Disks		

Peripheral Devices required:

Terminals _____

Graphics Equipment _____

Printers _____

Other _____

Other software required to execute 4GL:

_____

Does the 4GL have communication capabilities? _____

Can the 4GL interface directly with other software packages?

Spreadsheets _____
Word Processing _____
Statistical Analysis _____
Financial Modeling _____
Project Management _____

Does the 4GL provide backup and recovery features? _____

Does the 4GL support a multi-user environment? _____

ADDITIONAL QUESTIONS

Does the 4GL support network pointers or provide networking capabilities?

Can the 4GL interface directly with:
- statistical analysis packages
- financial modeling packages
- project management packages

Can user profiles be programmed?
Are self-explanatory system messages provided?
Does the 4GL handle system defaults?
Will the system warn the user if the desired process will consume hardware resources or take a long time?

Is there protection of data during abortion or system crash?
Will the system restart automatically?
Are there automatic system restarts at the task level?
Is data that has been involved in a crash identified?

Can users store and catalog user procedures?
Can users save processing results in files for later processing?
Can users create new files or allocate storage space from within the 4GL?
Can the 4GL window files or functions?
Can users create a library or catalog of files and functions?

3.2 USER INTERFACES

There are many different kinds of user interfaces to 4GL which provide many different functions. Some 4GL systems are mainly menu driven while others utilize "natural English" commands. Others employ graphic symbols as input and output. The kinds of interfaces that are appropriate for a given application are determined by the kinds of users that will be utilizing the system, the level of user expertise, etc. **A Functional Model for Fourth Generation Languages** [FISH86] describes user functions as follows:

> *"User functions define those capabilities and services provided by a 4GL to address the interaction between system users and the 4GL. These functions define a high level dialogue management capability in the sense that much of the housekeeping concerning the interaction between the 4GL and the user is managed and performed by the 4GL. This area is further broken down into the following specific functions:*
>
> - *Screen formatting*
> - *Menu management*

o Message prompting
o Logical device management for devices such as light pen, touch-screen, mouse, graphics tablet, remote sensors, etc."

For selection consideration these functions can be viewed as features or interfaces that a product may possess. Some experts suggest identifying whether or not a system is user-friendly as a selection criteria but user-friendly is a somewhat nebulous term. There are more objective criteria that can be used. For example, does the system have on-line help, menus, defaults for novice users, the ability to tolerate typographic errors, and the ability to provide alternative forms of commands to correct "syntax errors"? If the system had all of the above, it might be considered user-friendly for novice users.

Menus and defaults might be features that experienced users would want to avoid. These features would hinder work rather than provide necessary assistance. Sometimes systems have a novice and an expert mode or an interface with different levels of verboseness. If many kinds of users will be using the 4GL, multiple levels of interface would be very important.

If the 4GL application requires any special kind of interface such as a touch-screen, mechanical mouse, or remote sensor, the availability of these devices with the 4GL needs to be determined and a detailed analysis conducted on how the device is interfaced. Special device interfaces are not all that common with 4GLs at this time so it is an area in which to beware of vaporware, or the "to be delivered in the near future" syndrome.

Professional programmers may want the capability of designing screens to serve as a user interface. If so, the 4GL should provide the following screen functions; field character validation, field masks, required fields, filled fields, video intensity control, computed fields, default field values, screen/data field refresh, inter-field checks, and inter-record checks.

Some systems have added an artificial intelligence (AI) or expert system interface. In these systems, most of the language of the 4GL is hidden from the user by the AI or expert system. These AI systems act as the user interface for the 4GL. There are few of these systems available currently but the number is increasing. If the application justifies an AI interface, these characteristics need to be analyzed very carefully because most systems currently available are very specialized.

USER INTERFACE CHECKLIST

SCREENING INFORMATION

Does the 4GL provide menus? _____ Screens? _____

Does the 4GL have novice and expert modes? _____

Is there substantial on-line help available? _____

Is the 4GL mainly intended for:

professional programmers? _____
non data processing end-users? _____

ADDITIONAL QUESTIONS

Does the 4GL provide "natural" English?
Does the 4GL have defaults to help novice users?
Can an expert user override the defaults?
Is the 4GL tolerant of typographic entry mistakes?
Are there help messages available?
Are there different levels of verboseness?
Is on-line documentation available?
Is there a split-screen capability?
Will the 4GL suggest corrections to syntax errors?
Can there be different user's views of data?
Will the 4GL prompt the user for responses?

Is there full-screen cursor positioning?
Is there graphic symbol manipulation?
Can users develop input screens? output screens?
Is there a facility for scrolled questions and answers?

Is there a logical device facility so that the 4GL can control:
light pens, voice input-output, sensors, etc.?

Does the 4GL have an AI or expert system interface?

3.3 LANGUAGE FEATURES

There are a number of language functions involved in a 4GL, query language functions, programming or command language functions, the function of the language the 4GL is written in and the function of interfacing with other programming languages. All four are very important to the operation and functionality of the 4GL.

3.3.1 Portability and Performance

The language the 4GL is written in affects the potential portability of the 4GL. If the 4GL is written in assembly language code it probably will not be as portable to another system as a system written in a standard programming language. There has been very little documented experience with porting a 4GL so no presumptions should be made. Most 4GLs are closely tied to operating systems which raises additional problems for portability.

Some 4GLs have an interpretive mode and a compile mode. That is, the 4GL can run interpretively for development work, ad-hoc queries, prototyping, etc., but can generate and compile the source code when a project has been completed and is in production. Generally the compiled code decreases the runtime and produces an optimized mode of operation. If the 4GL produces source code in a standard programming language that can be compiled, it may be more feasible to port this to another system.

Systems which are hardware resource consumptive will be less of a problem in a 4GL which has an optimizing capability. For any kind of production system, or application which is going to be run repetitively, an optimized mode should be a mandatory selection criteria. Applications such as ad-hoc queries, forecasting, and decision-making will probably not be able to use this mode.

3.3.2 External Language Interface

Another function of the 4GL is interfacing with another programming language, usually a third generation language such as FORTRAN, COBOL, or BASIC. Some 4GLs currently available have a third generation programming language interface. Most applications should require this capability. Most computer installations have a large number of programs or a body of code that already exists. In many cases, most of that code will still need to be utilized, and probably needs to be interfaced with the 4GL. All third generation code is not rewritten overnight in a 4GL, and much of it should not be rewritten.

There are probably a number of programs that will need to be run with the 4GL. For example, if there are routines which handle security, these need to be able to interface with the 4GL. The 4GL may need to call routines to handle graphics, off-line printing, statistical calculations or many other computer applications that are currently running. If the 4GL does not have the capability to interface with the languages that these routines were written in, it will not be able to directly control these applications and will make integration of the 4GL much more difficult.

One other reason a programming language interface for a 4GL is important is that the 4GL may not address all situations for the user's environment. Without the language interface, writing a subroutine in a third generation language to handle such situations may be impossible as may interfacing with another 4GL. Since few systems, if any, can accomodate all possible conditions for all users, reality dictates the inclusion of this programming language interface feature as a selection criteria.

3.3.3 Programming Language

Most 4GLs have some kind of programming language, command language or procedural language in addition to a relatively non-procedural end-user or query language. This is called a "professional/technical user language" in [FISH86] and is described as follows:

> *"Many Fourth Generation Languages available today include a more comprehensive language for use by professional programmers in creating extensive applications with the 4GL. The constructs available in this language are not application specific and usually require much more technical expertise in application evolution than is observed in most 4GL end-users.*
>
> *The capabilities of this language include mechanisms for manipulating virtually all components of the 4GL. It is here that many 4GL vendors implement the commands necessary to perform screen formatting, report generation, and procedure definition."*

Features relating to the procedural language that should be considered include, whether command files can be established to run for other users or to speed up certain processes, whether the language has all the capabilities required by the application, and whether this procedural language is fully integrated to all the other features of the 4GL or limited to certain functions.

3.3.4 End-User Language

The non-procedural language is called an "end-user language" in [FISH86] and is described as follows:

> *"Part of the meaning of language in the term 'Fourth Generation Language' is based on the existence of a programming language designed for use specifically by 4GL end-users. ... Typically, an end-user will not differentiate between the command language used to operate a 4GL and the language used to execute reports, display screens, and define procedures.*

Examples of the types of commands found in end-user languages are - -

- *o COMPUTE, ADD, and SUM for performing numerical computations;*
- *o SELECT, JOIN, and DISPLAY for retrieving and presenting data in a relatively straightforward form such as would be found in many query components; and*
- *o PRINT, SUBTOTAL, and TABLE LOOKUP for retrieving and presenting data in tabular formatted form such as in printed reports." (see Section 3.5 "Report Writing Features")*

The syntax of this non-procedural language needs to be analyzed to see if it is appropriate for the kinds of users of the application. Some systems provide assistance with syntax errors, such as repeating the users' input with suggestions for other syntax or fairly complete error diagnostics that will not only tell the user they have just gotten "error 560" but that this error means they have forgotten the punctuation for that command. Assistance like this is important to novice users or professional users who use the system infrequently.

Some vendors have built a non-procedural language implementation on a query language foundation. Some of the systems utilize a proposed standard query language for relational systems, SQL, as part of the 4GL. Since much of the detail of the query language relates to the capabilities of the database, many of the query language features will be discussed in the section on data management.

LANGUAGE FEATURES CHECKLIST

SCREENING INFORMATION

Is there a non-procedural user language? _____

Is there a procedural, command, or programming language? _____

Can users search using Boolean Logic and logical comparison operators (AND, OR, ‹, ›, =, NOT =)?_____

Can the 4GL interface with other programming languages directly?_____
Does the 4GL produce source code? _____
If so, what programming language? _____
Does the 4GL produce compiled or optimized code? _____

ADDITIONAL QUESTIONS

Is the user language a professional/technical oriented user language?
Is the user language a novice oriented language?

Can command or programming files be established? saved?
Can the system handle parameter lists at runtime?

Are the components of the 4GL fully integrated or do the users have to exit from one to enter another (for example exit the query section to enter the report writer)?
Is there a different language or syntax for different components of the 4GL?
Are the editing functions integrated or is there a different editor for program or command files than for data?

Does the 4GL utilize SQL or NDL?

Are defaults provided to assist users?
Is there syntax correction or assistance?

Can the system handle Boolean logic and logical comparison operators for searching or selecting (AND, OR, GREATER THAN, LESS THAN, EQUAL TO, NOT EQUAL TO)?
Can the users nest operators for searching or selecting?

Can the users use commands such as COMPUTE, ADD, and SUM for performing numerical computations?
Can users SELECT, JOIN, and DISPLAY data for retrieval and presentation?

Can the system handle:
mathematical functions?
financial functions?
statistical functions?
logical functions?

3.4 SECURITY FEATURES

The security features of a 4GL often encompass password protection at various levels, backup protection, encryption capabilities, and audit trails. The level of security required can only be determined by the application.

A minimal requirement for 4GL implementations should be some facility for automatic backups and protection of data during system crashes. The user or support personnel should be able to backup the entire system, database, program files, screens, etc. at least weekly and may want to supplement this with daily

incremental backups. Micro-based systems do not often offer the extent of backup facility that mini- or mainframe-based systems do. The backup features may protect the data from being accidently destroyed but do not provide any kind of protection for the data usage.

Password protection can exist on the system, user, file, record, table, or data item level. System implementations of passwords vary and are sometimes dependent on the operating system. There are many ways the passwords can be implemented. For example, some systems scramble passwords or require that the passwords be changed after a certain amount of time.

Some system implementations incorporate encryption capabilities with the 4GL. The implementations can vary with encryption being used at various levels like the password capabilities.

Some systems provide various kinds of auditing capabilities such as identifying users who access various parts of the system, identifying the kinds of actions taken by various users, analysis of the data utilized by users, and analysis of the interactions between users.

Another approach to security is to provide some kind of security, password, encryption, audit, etc. at the command level. This means that perhaps only certain users can write to a file or read certain files, join certain files, etc. Usually this command level security is linked to the users identification but may be linked to certain applications. For example, a specific user may be able to access payroll information for running a payroll accounting application but may not be able to access the payroll information when performing certain personnel actions such as EEO (Equal Employment Opportunity) analysis.

All of these security attributes need to be analyzed in great detail if security is a major concern. In some cases the security may be provided by the operating system or by programs already existing on the system. In these cases the 4GL must be able to access the operating system features or have a programming language interface to run the security programs. The database structures of the 4GL may also have to conform to requirements of the security operation. It is important to insure that using the 4GL does not bypass other security measures, i.e. that using the 4GL does not provide "back door" access to database systems, etc. There needs to be a consistant treatment of access privileges for the operating system, the 4GL, the databases, etc.

The Institute for Computer Sciences and Technology has produced numerous publications on computer security. This report does not attempt to repeat all of the information and guidelines that

would be relevant. For further information users should refer especially to the following Federal Information Processing Standards (FIPS) Publications:

[FIPS74] **Guidelines for Implementing and Using the NBS Data Encryption Standard**;

[FIPS73] **Guidelines for Security of Computer Applications**;

[FIPS65] **Guideline for Automated Data Processing Risk Analysis**;

[FIPS87] **Guidelines for ADP Contingency Planning**;

[FIPS102] **Guidelines for Computer Security Certification and Accreditation**; and

[FIPS112] **Password Usage**.

Users may also want to refer to:

NBS SP 500-33 **Considerations in the Selection of Security Measures for Automatic Data Processing**, Michael Orcyre and Robert H. Courtney, Jr., Gloria R. Bolotsky (Ed.), June 1978. [ORCY78]

NBS SP 500-134 **Guide on Selecting ADP Backup Processing Alternatives**, Irene E. Isaac, Nov.1985. [ISAA85]

SECURITY CHECKLIST

SCREENING INFORMATION

Is the security based on password protection? _____

Are there additional security features such as scrambled passwords, encryption, and audit trails? _____

ADDITIONAL QUESTIONS

Is there password protection at the following:

- system level?
- user level?
- database level?
- file level?
- record level?
- data item level?

Is there automatic backup protection?

Is there an encryption capability?
    If so, what level of encryption is provided?

Is there separate protection for the data dictionary?

Does the system track and report unauthorized attempts to enter
        the system?
Are logs of usage maintained at the system level?

Is there security based on scrambled passwords?
Are there levels of security based on the command type?

3.5 REPORT WRITING FEATURES

A report presents the results of some process. Reports to the screen usually have fairly simple characteristics. Reports that are generated by queries, tabulations, etc. and printed on paper, viewgraphs, stationery, forms, etc. generally have more complex characteristics. Quite often these report characteristics have both hardware and software requirements.

There are many very specific questions that can be asked about the reporting capabilities of 4GL if this area is very important to the application. Some vendor implementations group all of these functions in a separate report writer while others handle some functions via a query language and a report writer. In some systems the data dictionary can issue additional reports about the database structures. For selection criteria purposes it does not matter how these functions are implemented but simply whether or not they are available.

If hard copy reports are important to the application, the availability of features such as zero suppression, adding commas to number fields, column titles, row titles, floating dollar signs, page breaks, footnotes, headings, pagination, and multiple fonts is important. If the reports are to be printed on screen, then screen formatting capabilities are important.

For some reports certain mathematical functions may be important such as the ability to produce totals, sub-totals, log functions, accumulations, and counters. If the data from these reports are going to be presented in graphic form, the 4GL must have an interface from the report processor to the graphics processor. There are a number of systems that will let users present data in graphic form but not directly from the report generator. The data in these systems has to be rekeyed as data entry to the graphics system. If the application requires a substantial amount of data transfer from the report processor to the graphics processor this needs to be implemented directly by the 4GL with no rekeying.

Different users require different report writing capabilities. For example, novice users will want to be able to generate 'quick and dirty' reports without having to tell the 4GL where to put each item of information. They will need to have access to defaults that can be utilized for formatting, and viewing or printing the reports. They may want to have the ability to retrieve column headings from existing tables or from the data dictionary. As these users get more knowledgable they may want to be able to specify some special features so they will need to be able to override these defaults.

If there are special report writing features such as the ability to process forms, stationary, typesetting, etc. the report writing interfaces to the 4GL will have to be analyzed carefully. The system may also need to access existing programs such as those for typesetting so a programming language interface may be required. Either the 4GL or the system or some other programs must be able to drive the printing devices. Thus either the 4GL must have the driver capability or it must have the interface to the drivers.

REPORT WRITING FEATURES CHECKLIST

SCREENING INFORMATION

Are there defaults for simple reports? _____

Are there capabilities for more complex, user specified reports? _____

ADDITIONAL QUESTIONS

Does it handle:
- sub-totals?
- column totals?
- row totals?
- percent calculation across columns?
- percent calculation down columns?
- accumulations?

Does it allow:
- user defined reports by data item?
- zero suppression?
- floating dollar signs?
- adding commas to number fields?
- user specified spacing?
- user specified page breaks?
- table lookup?

printing odd size printouts?
row titles?

Does it provide?
footnotes?
more than one layer of headings?
automatic pagination?
automatic spacing?
processing multiple reports in one pass?

Can users print on forms or stationary?

Does it allow a number of fonts to be specified and accessed?
Can users specify the number of records to be processed for a separate report?

Is the report output independent of the type of printer used?

3.6 DATA MANAGEMENT FEATURES

Data management functions are described in **A Functional Model for Fourth Generation Languages** [FISH86] as follows:

> *"The area of data management in the context of 4GL includes capabilities necessary to describe data structures, store and retrieve instances of data, and provide facilities to secure the content and integrity of the data. Functions in this area include the following:*
>
> - *o Logical data structure management*
> - *o Data storage and retrieval*
> - *o Archiving and restoration*
> - *o Auditing*
> - *o Data security"*

The vendor implementations of these functions vary greatly. Many vendors would implement these functions in a query language, a data dictionary or data definition facility, and a database system. For selection criteria it does not matter what the implementation is but simply whether or not certain functions are available. Because security should be such an important factor in the selection process, archiving and restoration, auditing, and data security are discussed in detail in Section 3.4.

For 4GL implementations that have a data dictionary system, the first selection question is whether or not the data dictionary is active or passive. There are many definitions of active and passive data dictionary systems and systems with varying degrees of these two states. International Data Corporation defines these systems [INTE86] as follows:

"Passive: These products passively store data descriptions. They do not interact with or control other software components within the processor. They serve only as reference for the descriptions which they store.

Active/Integrated: These products interact with the database manager and most likely with other software tools such as report writers, query languages, fourth generation languages, etc., in order to drive, control, and monitor these software tools. The degree of activity depends on the manner in which the metadata . . . is bound to the tool tool which is to use its metadata."

Some systems have a data definition facility which acts like a data dictionary. It is relevant to determine if this is active or passive also. Sometimes the data dictionary or data definition facility can provide facilities for table lookups to other parts of the 4GL, such as, column headings to reports, labels for graphs, synonyms for queries and reports, and cross references for programs. Sometimes the data dictionary drives the report generator or the query facility. Some data definition or data dictionary systems store synonyms and aliases of data element names and user remarks.

The requirements for data dictionary functions are determined by the size and scope of the application. There is a substantial standards effort in this area manifested in a draft proposed American National Standard [ANSI86e] and the U.S. Federal Information Processing Standard Information Resource Dictionary System. Currently no known 4GL implementations include the proposed data dictionary standard. These should be appearing in the near future. When these become available, it would be important to ask if the data dictionary facility meets the standard, especially if there are other programs on the system utilizing this standard. Further details on the proposed standard Information Resource Dictionary System are available in [GOLD85].

Some of the functions of a query language are discussed in Section 3.3, "Language Features." However, many of the features of a query language are dependent on the structure of the database and therefore closely tied to the data management functions. There have been extensive standards efforts on query languages including a proposed standard query language for a relational database, SQL [ANSI86c], and a proposed standard query language for a networked database, NDL [ANSI86b]. It is important to know if the vendor's implementation addresses either of these proposed standards. Decision makers should know what the structure of the database is, if it is relational, hierarchical, or networked or even quasi-relational, to help determine the appropriateness of the application to the language and the

database. Other functions such as joining files, searching records, tables or files are functions of both the database and the query language. Some applications are appropriate for relational systems while others may not be.

There are certain characteristics of the database functions that must be learned such as the maximum limits on files, records, fields per record, characters per field and tables. Sometimes this is very difficult because the facts are not clearly stated. Often the maximum limit of one means that the stated maximum for some of the others cannot be utilized. For example, if the fields are utilizing the maximum character length, the number of tables may be limited. The maximum file size may limit the number of records if the records are at the maximum length. The total memory available is the principal limiting factor.

DATA MANAGEMENT FEATURES CHECKLIST

SCREENING INFORMATION

Is there a data dictionary or data definition facility? _____
If so, is there an active data dictionary? _____
Can users relate or JOIN files? _____
What is the maximum number of tables or files that can be joined? _____

Does the system allow text or variable-length records?

Will the database accept foreign files and create files for foreign systems? _____

What is the maximum allowable size of:

files? _____
records? _____
fields? _____
fields per record? _____
characters per field? _____
tables? _____

ADDITIONAL QUESTIONS

Does the data dictionary or data definition facility provide:

table lookups?
column headings to reports?
data fields to be referenced by field name or synonym?
cross reference usage tables?
file descriptions?

Can the data dictionary or data definition facility be customized
with user remarks?
Can attributes and tables be defined quickly and easily?
Can new tables be added using existing attributes?

What model does the database most closely match:
relational?
hierarchical?
networked?
quasi-relational?

Can the user access data by:
any field?
any portion of the data (mask)?
any character strings?
phonetic equivalents?
exclusion/inclusion from a set?
specified ranges?

Can the user generate indexes?
Can secondary indexes be added?
Can the user specify kinds of searches (binary, sequential, etc.)?
Can the user add or delete fields, columns, or records to the
database?
Can users perform sorts?
ascending?
descending?
maximum number of fields or records _____
only on indexes?
Can users merge results?

Will the system accept data from external formats?
Will the system convert data to external formats?
Can data files be reorganized without reentering or loading files?
Can data be entered in batches (not record by record)?

Does the system allow variable-length records and fields?
Can the system handle text?
Can the system handle multidimensional arrays or matrices?
Does the system allow concurrent access to the database?

Can variable data be entered into procedures:
by prompting the user?
by specification at runtime?
by system substitution?

Can input to the system be edited and verified for:
proper format?
proper range?
match of all keys?
table lookup?

Is an audit trail or transaction record kept for:
- all transactions?
- specified transactions?
- rejected transactions?
- all valid transactions?

Does the system handle arithmetic and logarithmic functions?
Does the system allow concatenation of data strings?
Can the user change the sequence of computations?

Can the system test for equal or not equal to?
Can the system test for <,>, or <=, >=?
Can the system test for Boolean combinations therof?
Can the system select on substrings?

Can the database handle first occurence/last occurence?
Can the database handle minimum/maximum?

3.7 GRAPHICS FEATURES

Many 4GLs now have graphics facilities integrated with the other functions of the 4GL. If graphics facilities are of major importance to the application, the decision-makers need to analyze very closely any specialized equipment required for this use and how the 4GL fits into any existing graphics environment. For example, if the 4GL does not have the ability to call another programming language, users may not be able to run currently installed graphics subroutines. It may not be sufficient for the 4GL to provide some of the capabilities needed because the users may not easily be able to augment these with existing capabilities.

There is a wide range of graphics capabilities currently available. Some applications only require simple business graphics; bar charts, pie charts, and line graphs. Other applications may require three-dimensional graphs, scatter diagrams, and logarithmic graphs. There are a number of 4GLs that can provide business graphs or allow interfacing to a spreadsheet facility that provide these graphics capabilities. There are few 4GLs that can handle the more complex graphics requirements. For those applications that require complex graphics, the availability of 4GL interfaces to other languages, graphics software, and peripheral devices is critical.

The most important feature of the graphics capabilities is its integration with the other functions of the 4GL. The amount of labor required for creating a graph whose data points are obtained directly from the database and whose labels are obtained from the data dictionary is much less than for graphic

implementations requiring re-entry of this data. This characteristic is not always readily apparent from sales literature or vendor presentations.

Novice users may need the graphics features to perform scaling, calculating the percentages for pie chart, placing labels, formatting bar placement, etc. They may want the ability to view the chart on the screen and manipulate the results before any printing takes place. If the users want high-quality presentation graphics, they may require the 4GL to support high-resolution screens, high-resolution printing devices, color, text capabilities and multiple font selection.

One important aspect the decision-makers may want to consider if high-quality graphics or complex graphics capabilities are desired is whether or not the graphics facilities support the first Federal graphics standard, FIPS 120, **Graphical Kernel System (GKS)**, April 18, 1986, and other federal, national and international standards that are in various stages of development. References to these additional standards are given below.

> **Programmers Hierarchical Interactive Graphics System (PHIGS)**, document dpANSI X3.144, X3 Secretariat, CBEMA, Washington, D.C., 1986. [ANSI86f]
>
> **Graphical Kernel System (GKS) Functional Description**, ISO 7942 and ANS X3.124-1985, ANSI, New York, NY, 1985. [ANSI85b]
>
> **Graphical Kernel System for Three Dimensions (GKS-3D)**, ISO DP8805, ANSI, New York, NY, 1986. [ANSI86d]
>
> **Computer Graphics Metafile (CGM)**, dpANS X3.122-1986, X3 Secretariat, CBEMA, Washington, D.C., 1985. [ANSI85a]
>
> **Computer Graphics Interfacing Techniques for Dialogues with Graphical Devices (CGI)**, document X3H3/85-173, X3 Secretariat, CBEMA, Washington, D.C. 1986. [ANSI86a]

Readers may also want to refer to [IEEE86], [BRAN86], [BONO85], and [ABIE86] for general discussions of the graphics standards.

These application programmer interface standards and graphics device interface standards are important for providing device independence for the users.

GRAPHICS FEATURES CHECKLIST

SCREENING INFORMATION

Are there graphics capabilities for simple business graphics? _____

Are there capabilities for handling complex graphics? _____

ADDITIONAL QUESTIONS

Is there a graphics generator?

Are there facilities for producing:
- line graphs?
- bar charts?
- pie charts?
- scatter plots?
- 3-dimensional graphs?

Are there defaults automatically provided for the line graphs, bar charts and pie charts?

Does the graphics facility use table lookup?

Does the graphics facility use the data dictionary for labeling?

Does the graphics facility allow utilization of arithmetic and logarithmic functions?

Is there a facility for character graphics?

Is there a facility for different fonts?

Is there a facility for alphamosaic graphics?

Does the graphics facility support vector graphics?

Does it support raster graphics?

Does it support color graphics?

Can the user manipulate or rotate the generated charts?

Can the graphics facility link directly to the query language or report writer to obtain data, eliminating rekeying?

3.8 IMPLEMENTATION ISSUES

There are several different selection issues that relate to the implementation of the 4GL and the support for the 4GL. The first issue that should be considered is the quality and quantity of documentation for the 4GL. Some vendors provide very complete reference manuals, while others provide only very cryptic manuals. Some vendors put some of the documentation on-line with

the system as a separate entity while some provide short descriptions in the form of help messages. The scope of the application, type of users, and hardware implementation help to determine what kind of documentation is most appropriate.

Other selection considerations relating to documentation are the timeliness of the documentation and the cost. Some systems have documentation that is badly out of date and not updated very often. It is really difficult to run a system with documentation that does not match the current release being run. Other users of the system would be the best source of information on the status of the documentation. The cost of the documentation varies greatly. For example, some vendors of micro-based 4GLs do not provide the documentation with the system because they presume that users have the mainframe documentation. Thus, the micro documentation is only available for an additional fee.

The kinds of support offered by vendors also vary and are difficult to assess without talking with other users. The costs associated with the support are sometimes difficult to identify at the time of the sale. Adequate support for an organization with no in-house facilities can mean maintenance for the 4GL, training provided by the vendor, and a hot-line support facility. Other organizations may be able to provide support for the 4GL with in-house personnel. Decision makers need to judge the kinds of support that will be necessary and look to see if the vendors can provide it.

The cost of the 4GL may be one of the selection criteria. This cost is not just the purchase, lease or rental price of the 4GL but also the costs associated with support, maintenance, and training for the system. Personnel costs and the amount of training required may have to be estimated to establish an accurate figure of the 4GL costs.

Another selection criterian should be the stability and viability of the vendor, especially if there is very little in-house support available for the system. The length of time the vendor has been in business, the size of the company, the number of installations of the software, and the financial posture of the vendor can be considerations. The length of time the 4GL has been released and number of users also indicate the status of acceptance and user base that might be available for personnel or additional information. Quite often brand new software is released with a fair amount of "bugs". By the time the software has been through its second and third release, many of these "bugs" have been corrected. Again, this is more important to installations that have very little in-house support available.

IMPLEMENTATION ISSUES CHECKLIST

SCREENING INFORMATION

Does the vendor provide - training? _____
maintenance? _____
hot-line support? _____
on-line documentation? _____
reference manuals? _____

What is the current number of installations? _____

What is the current pricing structure?

purchase price _____

rental or lease _____

How long has the 4GL been on the market? _____

ADDITIONAL QUESTIONS

Is the user documentation easy to follow?
Is there an: installation manual?
reference manual?
users guide?
summary of commands?
Will the vendor customize the system?
Will the most current releases be distributed automatically?
Is some training provided with the purchase price?
Is some kind of hot-line provided for no additional price?
If so, is it a toll free number?
If so, are you able to get through to the number?

4.0 RECOMMENDATIONS ON 4GL USE

In the introduction to this report, four areas of concern were listed for 4GL usage:

- o Performance - hardware resource consumption, response times, multi-user access, real-time processing
- o Portability - no 4GL standards, portability of language, portability of developed code, number of skilled programmers, number of revisions and updates
- o Support - training, maintenance, transferability of skills, vendor support
- o Relevance - applicability to organization, problem or application, and methodology for software development

Some of these areas of concern have been discussed in the sections on specific features, such as portability and performance in Section 3.3. This section will cover two remaining areas related to these concerns; how 4GLs are used, and what 4GLs are used for. The first area, how 4GLs are used, is a study of the data processing environment and management of the software development process. The second area, what 4GLs are used for, is a study of the applications that are appropriate for 4GL usage. The final section, "Factors For Success," also relates to some of the concerns listed above such as training and support but includes recommendations for facilitating a successful 4GL implementation.

4.1 4GL USAGE

4GLs are changing the approach to software development and the management of data processing resources. Traditional environments required that most software development be completed in the data processing department by computer professionals. Users would only be involved in reviewing written specifications and approving iterations of these specifications. This approach to software development produced a backlog of applications development waiting for completion and many disgruntled users.

In a 4GL environment, the user has a dynamic role in the software development process. In some organizations the users are developing small applications themselves with 4GLs. For larger applications, the users are at least able to review working prototypes of the proposed system and are encouraged to make changes during the prototyping process. In either case, the role of the user has been dramatically changed.

The use of prototyping with a 4GL has altered the traditional software development life cyle. Prototyping presents a relatively inexpensive and quick method of developing and testing an application system. Prototyping requires the user and the data processing professional to collaborate on the development of the application. The prototype (in the 4GL sense) is an actual working system, not a simulation or specification on paper. Prototypes can be used to test assumptions about the user's requirements and the system design. This is an interactive, iterative process with prototypes being altered as additional functions and suggested changes are made to the system. Prototyping is an evolutionary process that does not distinguish between development and maintenance.

Prototyping and fourth generation languages will be more fully described in a future report. There are also a number of articles relevant to these areas such as [BABC86], [BOAR84], [CULL85], [EDPA84], [KANI85], [LANT86], [MART85], [MESS84], and [SCHA84]. Readers may want to refer to these for further information.

4.2 4GL APPLICATIONS

4GLs are not by design, "one size fits all." Just as there are 3GLs that are more appropriate for some applications, for example, scientific languages for scientific applications and simulation languages for modelling and simulation, there are 4GLs that are designed to handle specific applications. Some 4GLs are designed to execute primarily ad-hoc queries and small interactive programs for business applications. Other 4GLs are designed for facilitating the development of large application software systems. The first type of 4GL is appropriate for end-user computing or use in an information center environment. The second type is much more appropriate for a traditional programming shop.

Since one of the advantages of a 4GL is that housekeeping chores of programming are automatically hidden from the users, certain file operations, bit manipulations, etc., are not readily available. Thus, applications which require extensive data manipulation and computation, such as "refinery operation, satellite image processing, air traffic control, or rocket launches," [MART85] may be inappropriate in a 4GL. The more complex an application, the more difficult it is to use a 4GL for its solution. There are some 4GL products which are appropriate for program generation and heavy-duty computing but most are not. "4GLs are more commonly used in decision-support systems and management information systems than in the development of routine DP." [MART85]

In general, 4GLs seem most appropriate for applications such as:

- o decision-support systems,
- o management information systems,
- o end-user computing,
- o data summary and simulation,
- o ad hoc queries and reports,
- o short-term applications or jobs which would be utilized at infrequent intervals, and
- o prototyping tasks.

4.3 FACTORS FOR SUCCESS

There are numerous factors involved in the successful implementation and use of any software system, including the selection and acquisition of an appropriate system. For 4GLs there are three requirements that are especially critical for success:

- o the development of a comprehensive organizational plan for incorporating a 4GL,
- o training and support for 4GL users, and
- o the establishment of procedures and methodologies for using 4GL.

If the selection process has been followed correctly, a 4GL has been selected which should be appropriate for the structure of the organization, the hardware and software environments of the organization, and the type of organizational users. Before the 4GL is installed, a plan must be developed for implementing the 4GL and incorporating the 4GL into the existing computing facilities. This plan needs to address training and support, the policies for 4GL use, appropriate implementation milestones, and procedures for monitoring the problems and successes of 4GL usage.

The type and amount of training and support needed can only be determined by analyzing the applications for the 4GL and reviewing the assessment of the user environment (step two of the selection process). If adequate training and support are not provided, the users will not be able to utilize the potential of the 4GL. In addition, incorrect usage may cause hardware resource problems, software problems, and data problems which may impact all users of the computer facility. Management approval for training and support should be obtained before the 4GL is procured. In many cases, training should be established and a support organization formed before the 4GL is implemented.

The last factor for success is the establishment of procedures and methodologies for the use of the 4GL. Some guidance needs to be given to all users on; incorporating the 4GL with the established software development methodologies, using the 4GL with proven structured analysis and development techniques, use of 4GL for prototyping, the procedures for data administration and maintenance, and internal standards and practices that should be followed for areas such as documentation and security.

The use of the 4GL cannot be viewed frivolously. It must be planned for, supported, and formally implemented with the rest of the computing environment. Success is achievable with 4GLs if a suitable 4GL is selected, implemented, supported and properly utilized for appropriate applications and computing environments.

REFERENCES

[ABIE86] Abi-Ezzi, Salem and Bunshaft, Albert, "An Implementer's View of PHIGS," IEEE Computer Graphics & Applications, February 1986, p.12-23.

[ANSI85a] American National Standards Institute, **Computer Graphics Metafile (CGM)**, dpANS X3.122-1986, X3 Secretariat, CBEMA, Washington, D.C., 1985.

[ANSI85b] American National Standards Institute, **Graphical Kernel System (GKS) Functional Description**, ISO 7942 and ANS X3.124-1985, ANSI, New York, NY, 1985.

[ANSI86a] American National Standards Institute, **Computer Graphics Interfacing Techniques for Dialogues with Graphical Devices (CGI)**, document X3H3/85-173, X3 Secretariat, CBEMA, Washington, D.C., 1986.

[ANSI86b] American National Standards Institute, **Database Language NDL**, ANS X3.133-1986, X3 Secretariat, CBEMA, Washington, D.C., 1986.

[ANSI86c] American National Standards Institute, **Database Language SQL**, dpANS X3.135-198x, X3 Secretariat, CBEMA, Washington, D.C., 1986.

[ANSI86d] American National Standards Institute, **Graphical Kernel System for Three Dimensions (GKS-3D)**, ISO DP8805, ANSI, New York, NY, 1986.

[ANSI86e] American National Standards Institute, **Information Resource Dictionary System**, dpANS X3.138-198x, X3 Secretariat, CBEMA, Washington, D.C., 1986.

[ANSI86f] American National Standards Institute, **Programmers Hierarchical Interactive Graphics System (PHIGS)**, document dpANS X3.144, X3 Secretariat, CBEMA, Washington, D.C., 1986.

[ATRE85a] Atre, Shaku, "The Information Center and Productivity Tools: Working in Harmony," Computerworld FOCUS, Vol.19 No.37A, September 18, 1985, p.16-20.

[ATRE85b] Atre, Shaku, "Will the Real Fourth-Generation Languages Please Stand Up?" Computerworld, Vol.XIX No.24, June 17, 1985, p.57, 70.

[BABC85] Babcock, Charles, "New Jersey Motorists in Software Jam," Computerworld, Vol.XIX No.39, September 30, 1985, p.1, 6.

[BABC86] Babcock, Charles, "Prototyping, ADS/Online Join to Triple Productivity," Computerworld, Vol.XX No.14, April 7, 1986, p.29-30.

[BENI84] Benigni, D. R., Yao, S. B. and Hevner, A. R., **A Guide to Performance Evaluation of Database Systems**, NTIS PB85-155794/WLI, NBS Special Pub. 500-118, National Bureau of Standards, Gaithersburg, MD, December 1984, 58p. 70 refs.

[BLAN84] Blanpied, Greg, "Using Fourth Generation Languages Well," Software News, Vol.4 No.7, July 1984, p.27-8.

[BOAR84] Boar, Bernard H., "Alleviating Common Concerns of Application Prototyping - the Experience Difference," Computerworld, Vol.XVIII No.22, May 28, 1984, Special Report on Software Productivity, p.SR66-67.

[BONO85] Bono, Peter R., "A Survey of Graphics Standards and Their Role in Information Interchange," Computer, October 1985, p.63-75.

[BRAN86] Brannigan, Mike, "GKS: The New Graphics Standard," Computer Language, Vol.3 No.5, May 1986, p.26-30.

[CAMP85] Campbell, Donald F., "Reducing the Applications Backlog with 4GLs," Journal of Information Systems Management, Fall 1985, p.8-13.

[CANT84a] Cantral, David, "Fourth-Generation Tools Promote End-User Involvement," Computerworld, Vol.XVIII No.29, July 16, 1984, p.50.

[CANT84b] Cantral, David, "Speaking the User's Language," Software News, Vol.4 No.7, July 1984, p.24.

[CHRI85] Christoff, Kurt A., "Building a Fourth Generation Environment," Datamation, Sept.15, 1985, p.118-120.

[COBB85] Cobb, Richard H., "In Praise of 4GLS," Datamation, Vol.31 No.14, July 15, 1985, p.90-96.

[CUGI84a] Cugini, John V., **Selection and Use of General-Purpose Programming Languages - Overview**, NBS Special Publication 500-117, Volume 1, National Bureau of Standards, Gaithersburg, MD, October 1984, 81p.

[CUGI84b] Cugini, John V., **Selection and Use of General-Purpose Programming Languages - Program Examples**, NBS Special Publication 500-117, Volume 2, National Bureau of Standards, Gaithersburg, MD, October 1984, 178p.

[CULL85] Cullum, Ronald L., "Iterative Development," Datamation, February 15, 1985, p.92-4, 96, 98.

[DATA85] Datapro Research Corporation, **Perspective: Fourth Generation Languages and Personal Computers**, Delran, NJ, March 1985, p.MC13-200-301 - MC13-200-304, 20 refs.

[DESM85] Desmond, John, "Language Barrier: The Fourth Generation at Work," Computerworld, Vol.XIX No.45, Nov.11, 1985, p.1, 55, 61-62.

[DUBI86] Dubien, Ronald, "Choosing a Flexible 4GL," Computerworld, VolXX No.13, March 31, 1986, p.42-43.

[EDPA84] "Fourth Generation Languages and Prototyping," EDP Analyzer Special Report, Canning Publications, Vista, CA, 1984, 32p.

[EDPA85] "Speeding Up Application Development," EDP Analyzer, Vol.23 No.4, April 1985, p.1-16.

[EDPA86] "A Programmer Productivity Controversy," EDP Analyzer, Vol.24 No.1, January 1986, p.1-14.

[FIPS65] Guideline for Automatic Data Processing Risk Analysis, Federal Information Processing Standards Publication 65 (FIPS PUB 65), National Bureau of Standards, Gaithersburg, MD, August 1, 1979.

[FIPS73] Guidelines for Security of Computer Applications, Federal Information Processing Standards Publication 73 (FIPS PUB 73), National Bureau of Standards, Gaithersburg, MD, June 30, 1980.

[FIPS74] **Guidelines for Implementing and Using the NBS Data Encryption Standard**, Federal Information Processing Standards Publication 74 (FIPS PUB 74), National Bureau of Standards, Gaithersburg, MD, April 1, 1981.

[FIPS87] **Guidelines for ADP Contingency Planning**, Federal Information Processing Standards Publication 87 (FIPS PUB 87), National Bureau of Standards, Gaithersburg, MD, March 27, 1981.

[FIPS102] **Guideline for Computer Security Certification and Accreditation**, Federal Information Processing Standards Publication 102 (FIPS PUB 102), National Bureau of Standards, Gaithersburg, MD, Sept. 27, 1983.

[FIPS112] **Password Usage**, Federal Information Processing Standards Publication 112 (FIPS PUB 112), National Bureau of Standards, Gaithersburg, MD, May 30, 1985.

[FISH86] Fisher, Gary E., **A Functional Model for Fourth Generation Languages**, NBS Special Publication 500-138, National Bureau of Standards, Gaithersburg, MD, June 1986, 28p. 23 refs.

[FLET84] Fletcher, George W., "Talking to Your Computer in English," Software News, Vol.4 No.7, July 1984, p.30.

[FREE86] Freedman, David H., "Programming Without Tears,"High Technology, April 1986, p.38-43, 45.

[GALL85] Gallant, John, "Compatibility, Integration Key to Info Center Tool Choice," Computerworld, Vol.XIX No.34, August 26, 1985, p.14.

[GILL84a] Gillin, Paul, "Job Worries Seen Stifling Fourth-Generation Language Use," Computerworld, Vol.XVIII No.22, May 28, 1984, p.15-16.

[GILL84b] Gillin, Paul, "Large Shop Finds 4GL Meets Programmer's Needs," Computerworld, Vol.XVIII No.21, May 21, 1984, p.18.

[GILL84c] Gillin, Paul, "T. Capers Jones on Life Without Programmers," Computerworld, Vol.XVIII No.22, May 28, 1984, Special Report on Software Productivity, p.SR3-6.

[GILL84d] Gillin, Paul, "The Fourth Generation Moves into the Mainstream: an Interview with FOCUS Creator Gerald Cohen," Computerworld, Vol.XVIII No.14, April 2, 1984, p.17.

[GOLD85] Goldfine, Alan and Patricia Konig, **A Technical Overview of the Information Resource Dictionary System**, NBSIR 85-3164, National Bureau of Standards, Gaithersburg, MD, April 1985, 134p.

[GRAN85] Grant, Fred J., "The Downside of 4GLS," Datamation, Vol.31 No.14, July 15, 1985, p.99-104.

[GRUB85] Gruber, William H., "Software Selection with Vision," Information Center, Vol.1 No.5, May 1985, p.30-36.

[HARE82] Harel, Elie C. and Ephraim R. McLean, **The Effects of Using a Nonprocedural Computer Language on Programmer Productivity**, Information Systems Working Paper #3-83, Computers and Information Systems Research Program, Graduate School of Management, University of California, Los Angeles, CA, Nov. 1982, 26p. 26 refs.

[HEFF84] Heffernan, Henry, "4th Generation Limitations Emerge," Government Computer News, Vol.3 No.13, Dec. 1984, p.70.

[HEFF85] Heffernan, Henry, "Trade-Offs Found in FOCUS, COBOL Test," Government Computer News, Vol.4 No.11, June 21, 1985.

[IEEE86] IEEE Computer Graphics and Applications, August 1986 IEEE Computer Society, Los Alamitos, CA.

[INFO86] The Information Center Institute, "Checklist for Evaluating Data Base Management Systems," Information Center, Vol.II No.2, February 1986, p.49-50.

[INTE84a] International Data Corporation, **Application Generators,** IDC#2545, Framingham, MA, October 1984, 48p.

[INTE84b] International Data Corporation, **Fourth Generation Languages: Information Generators to Meet Information Needs**, IDC#2563, Framingham, MA, October 1984, 41p. 23 refs.

[INTE84c] International Data Corporation, "Look Who's Using Fourth Generation Languages," Software Watch, Vol.1 No.11, November 1984, p.1-9.

[INTE84d] International Data Corporation, **New Programming Languages**, Research Memorandum IDC#2483, Framingham, MA, May 1984, 22p.

[INTE86] International Data Corporation, **Data Dictionaries: Knowledge-Bases for the Future**, IDC#2838, Framingham, MA, January 1986, 50p.

[ISAA85] Isaac, Irene E., **Guide on Selecting ADP Backup Processing Alternatives**, NBS Spec. Pub. 500-134, National Bureau of Standards, Gaithersburg, MD, November 1985.

[JARK85] Jarke, Matthias and Yannis Vassiliou, "A Framework for Choosing a Database Query Language", ACM Computing Surveys, Vol.17 No.3, Sept. 1985, p.313-340, 109 refs.

[JOHN86] Johnson, James, "Users Fine-Tune 4GL Strategies," Computerworld, Vol.XX No.13, March 31, 1986, p.65-66, 70-71.

[JONE86a] Jones, T. Capers, "Appropriate Tools, Methodologies Ease Development Effort," Computerworld, Vol.XX No.8, February 24, 1986, p.48-49.

[JONE86b] Jones, Capers, "How Not to Measure Programming Quality," Computerworld, Vol.XX No.3, January 20, 1986, p.73, 76-79, 82.

[KANI85] Kaniper, Carol A., "Prototyping - New Tools for Cutting Through the Applications Backlog," Computerworld FOCUS, Vol.19 No.37A, September 18, 1985, p.41,44.

[KELL85] Keller, Joanne, "Talking About the IC Portfolio," Information Center, Vol.1 No.5, May 1985, p.20-24.

[KOLO85a] Kolodziej, Stan, "Bringing DSS into Focus," Computerworld FOCUS, Vol.19 No.37A, September 18, 1985, p.47-49.

[KOLO85b] Kolodziej, Stan, "DBMS Systems: How Will the Newcomers Fare?" Computerworld FOCUS, Vol.19 No.37A, September 18, 1985, p.11-13.

[LANT86] Lantz, Kenneth, "The Prototyping Methodology: Designing Right the First Time," Computerworld, Vol.XX No.14, April 7, 1986, p.69-72.

[LUKA84] Lukac, Eugene G., "The Impact of a 4GL on Hardware Resources," Datamation, October 1, 1984, p.105-6, 108-9, 112, 114.

[MART82] Martin, James, **Applications Development Without Programmers**, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1982.

[MART85a] Martin, James, **Fourth-Generation Languages Vol.1 Principles**, Prentice-Hall, Inc. Englewood Cliffs, New Jersey, 1985, 420p.

[MART85b] Martin, James (Editorial Advisor) and Pieter R. Mimno (Ed.), **The James Martin Report on High Productivity Languages**, Technology Insight, Inc., Marblehead, MA, 2 vols., 1985.

[MART86] Martin, James and Al Hershey, "Software Engineering Depends on Information Engineering," Software News, Vol.6 No.3, March 1986, p.60-62.

[MESS84] Messerich, Pat, "Prototyping Gives Users More Enthusiasm," Computerworld, Vol.XVIII No.22, May 28, 1984, Special Report on Software Productivity, p.SR48.

[MIMN85a] Mimno, Pieter, "4GL: Power to the Users," Computerworld In Depth Report, Part One, April 8, 1985, p.ID19-23, ID26-28.

[MIMN85b] Mimno, Pieter, "4GL:Power from the Products," Computerworld In Depth Report, Part Two, April 15, 1985, p.ID1-7.

[MORI85] Morison, Robert, "4GLs vs. COBOL," Computerworld, Vol. XIX No.32, August 12, 1985, p.ID15-20.

[MORI86] Morison, Robert, "Buying a 4GL: Be Forewarned," Computerworld, Vol.XX No.23, June 9, 1986, p.31,33.

[NIER86] Nierenberg, Nicolas, "Pitfalls Can Plague Unwary Benchmarking Teams," Government Computer News, Vol.5 No.3, February 14, 1986, p.34-35.

[NIES86] Nies, Thomas N., **Software Requirements in Environments for Effective End-User Computing**, POSPP Serial#P-14-6, Profit Oriented Systems Planning Programs, January 1986.

[ORCY78] Orcyne, Michael, Robert H. Courtney, Jr., Gloria R. Bolotsky (Ed.), **Considerations in the Selection of Security Measures for Automatic Data Processing**, NBS Spec. Pub. 500-33, National Bureau of Standards, Gaithersburg, MD, June 1970.

[PARI85] Parikh, Girish, "The Forth-Generation Maintenance Gap," Information Center, Vol.1 No.9, September 1985, p.44-47.

[PARK84] Parks, Merle, "Fourth-Generation Tools Mount a Good Offensive," Computerworld, Vol.XVIII No.22, May 28, 1984, Special Report on Software Productivity, p.SR4-8.

[PRES84] Pressman, Roger S., "Software Engineering Targets Future Concerns," Computerworld, Vol.XVIII No.22, May 28, 1984, Special Report on Software Productivity, p.SR9-12.

[ROSS84] Rossell, Gary G., "So You've Chosen Your 4GL - Now Can You Cost Justify It?," Computerworld FOCUS, Vol.19 No.37A, September 18, 1985, p.35-36.

[RUDO84] Rudolph, Eberhard, "What is a Fourth-Generation Language?," Computerworld, Vol.XVIII No.32, August 6, 1984, p.43-44.

[SANT84] Santarelli, Mary-Beth, "What Kind of Aid? It Depends on Skill," Software News, Vol.4 No.7, July 1984, p.22,24.

[SCHA84] Scharer, Laura L., "General Problem-Solving Methods Will Ease Your Move to Prototyping, If You Use Them," Computerworld, Vol.XVIII No.22, May 28, 1984, Special Report on Software Productivity, p.SR30-31.

[SCHU84a] Schussel, George, "Fourth Generation Productivity Tools - a Shopping Guide for Software Consumers," Data Management, Vol.22 No.10, October 1984, p.42-46.

[SCHU84b] Schussel, George and James Davey, "Legend Holds True for Fourth-Generation Tools," Computerworld, Vol.XVIII No.22, May 28, 1984, Special Report on Software Productivity, p.SR2,6.

[SHEL85] Shelby, Ronald and Ronald Dubien, "Choosing Your First (or Second or Third) DBMS," Computerworld, Vol.XIX No.7, February 18, 1985, p.ID/7, ID/10-16.

[SHEP85] Sheppard, Charles L., **Guide for Selecting Microcomputer Data Management Software**, NBS Spec. Pub. 500-131, National Bureau of Standards, Gaithersburg, MD, October 1985, 59 p. 11 refs.

[SPEN85] Spencer, David, "Beyond C: Programming Languages Past, Present, Future," Unix/World, Vol.11 No.6, July 1985, p.26-27, 30-35, 23 refs.

[STAH86] Stahl, Bob, "Friendly Mainframe Software Guides Users Toward Productivity," Computerworld, Vol.XX No.5, February 3, 1986, p.53-56,58,60,64-66.

[SNYD84] Snyders, Jan. "In Search of a 4th Generation Language," Infosystems, October 1984, p.28-30, 32.

[TINN85] Tinnierello, Paul, "Making a Sensible 4GL Selection," Computer Decisions, Vol.17 No.15, July 30, 1985, p.40-41, 44-45.

[WARR85] Warren, Carl, "Latest Languages Link Data To Diverse Tasks," Mini-Micro Systems, Vol.XVIII No.11, August 1985, p. 135-142.

[WILS86] Wilson, Richard, "The 4GL Evaluation Team," Information Center, Vol.II No.2, February 1986, p.38-40,42,56.

[ZACH85] Zachman, William F., **Unicorns or Goats? Fourth Generation Languages, Artificial Intelligence, and Related Topics**, International Data Corporation, Office of Technology Assessment, Vol.3 No.6, July 1985, 7p.

APPENDIX A: SUMMARY CHECKLIST FOR APPLICATION ENVIRONMENT

HARDWARE ENVIRONMENT -

Processor _____

Memory available _____

Other peripheral equipment available -

printers _____

graphics devices _____

terminals _____

modems _____

Links to other computers or networks - _____

_____

Office Automation equipment _____

_____

Security requirements _____

_____

SOFTWARE ENVIRONMENT -

Operating System _____

Languages that will interface _____

_____

Programs that will interface _____

_____

Database systems _____

_____

Structure or model of system _____

Data Dictionary _____

File handling _____

Data that will be accessed -

size of files _____

size of records _____

size of tables _____

size of fields _____

type of data _____

ORGANIZATIONAL ENVIRONMENT - (select one)

Highly structured and controlled, hierarchical levels of management _____

Structured levels of management but matrix managment for projects _____

Participatory management with informal direct lines of communication _____

OTHER ENVIRONMENTAL FACTORS

Organization's data processing applications - (select one in each grouping)

Commercial _____
Scientific _____
Mix of both _____

Highly time critical _____
Fairly flexible run times _____

Routine, repitious reports _____
Ad-hoc, interactive requests _____

High volume transactions _____
Low volume transactions _____

USER ENVIRONMENT

Identify the kinds of users of the proposed system. A matrix such as the following may be useful. Check all that apply:

	Managerial Users	Operational Users	Application Specialists
Novice			
.			
.			
.			
Professional			

APPENDIX B: 4GL SCREENING FORM

OPERATING FEATURES

On what hardware does it execute? _____

_____

On what operating system does it execute? _____

_____

How much memory is required: **MINIMUM** **SUGGESTED**

Main Memory _____

Hard Disk _____

Floppy Disks _____

Peripheral Devices Required: _____

Terminals _____
Graphics Equipment _____
Printers _____
Other _____

Other Software Required to execute 4GL:

_____

Does the 4GL have communications capabilities? _____
Can the 4GL interface directly with other software packages?
Spreadsheets _____
Word Processing _____
Statistical Analysis _____
Financial Modeling _____
Project Management _____

Does the 4GL provide backup and recovery features? _____
Does the 4GL support a multi-user environment? _____

USER INTERFACE FEATURES

Does the 4GL provide menues? _____ and screens? _____
Does the 4GL have novice and expert modes? _____
Is there substantial on-line help available? _____
Is the 4GL mainly intended for:
professional programmers? _____
non data processing end-users? _____

LANGUAGE FEATURES

Is there a non-procedural user language? _____
Is there a procedural, command, or programming language? _____
Can the 4GL interface with other programming languages directly? _____
Does the 4GL produce source code in a standard programming language? _____
Does the 4GL produce compiled or optimized code? _____

DATA MANAGEMENT

Is there a data dictionary or data definition facility? _____
If so, s there an active data dictionary? _____
Can users search using Boolean Logic (AND, OR, ‹, ›, =, NOT =)? _____
Can users relate or JOIN files? _____
What is maximum of tables or files that can be joined? _____
Does the system allow text or variable-length records? _____
Will the database accept foreign files and create files for foreign systems? _____
What is the maximum allowable size of:
Files _____
Records _____
Fields _____
Fields per record _____
Characters per field _____
Tables _____

REPORT WRITING FEATURES

Are there defaults for simple reports? _____
Are there capabilities for more complex, user specified reports? _____

GRAPHICS

Are there graphics capabilities for simple business graphics? _____
Are there capabilities for handling complex graphics? _____

SECURITY

Is the security based on password protection? _____
Are there additional security features such as scrambled passwords, encryption, and audit trails? _____

SUPPORT

Does the vendor provide - training? _____
maintenance? _____
hot-line support? _____
on-line documentation? _____
reference manuals? _____
What is the current number of installations? _____
What is the current pricing structure?
purchase price _____
rental or lease _____
How long has the 4GL been on the market? _____

NBS-114A (REV. 2-8C)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions)	1. PUBLICATION OR REPORT NO. NBS/SP-500/143	2. Performing Organ. Report No.	3. Publication Date Sept. 1986

4. TITLE AND SUBTITLE

Computer Science and Technology:

Guide to the Selection and Use of Fourth Generation Language

5. AUTHOR(S)

Martha Mulford Gray

6. PERFORMING ORGANIZATION (If joint or other than NBS, see instructions)

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
~~WASHINGTON, DC 20234~~ Gaithersburg, MD 20899

7. Contract/Grant No.

8. Type of Report & Period Covered
Final

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (Street, City, State, ZIP)

National Bureau of Standards
Gaithersburg, MD 20899

10. SUPPLEMENTARY NOTES

Library of Congress Catalog Card Number 86-600582

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)

This report provides guidance on the selection process for Fourth Generation Languages (4GLs). It also gives a description of the features, functions and capabilities of 4GLs; and a brief discussion on the use of 4GLs. A ten step selection process is suggested: 1) describing the application; 2) analyzing the application environment; 3) deciding on selection approach; 4) defining requirements; 5) developing list of desired 4GL features; 6) rating desired features; 7) selecting candidate packages; 8) rating 4GLs; 9)analyzing top few in detail; and 10)selecting 4GL. Checklists are provided for screening 4GLs and analyzing the application environment.

12. KEY WORDS (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)

end user computing; Fourth Generation Languages; 4GLs; high level language; nonprocedural languages; selection process; selection criteria

13. AVAILABILITY

☒ Unlimited

☐ For Official Distribution. Do Not Release to NTIS

☒ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.

☐ Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES
65

15. Price

ANNOUNCEMENT OF NEW PUBLICATIONS ON COMPUTER SCIENCE & TECHNOLOGY

Superintendent of Documents,
Government Printing Office,
Washington, DC 20402

Dear Sir:

Please add my name to the announcement list of new publications to be issued in the series: National Bureau of Standards Special Publication 500-.

Name _____

Company _____

Address _____

City _____ State _____ Zip Code _____

(Notification key N-503)

NBS Technical Publications

Periodical

Journal of Research—The Journal of Research of the National Bureau of Standards reports NBS research and development in those disciplines of the physical and engineering sciences in which the Bureau is active. These include physics, chemistry, engineering, mathematics, and computer sciences. Papers cover a broad range of subjects, with major emphasis on measurement methodology and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Bureau's technical and scientific programs. Issued six times a year.

Nonperiodicals

Monographs—Major contributions to the technical literature on various subjects related to the Bureau's scientific and technical activities.

Handbooks—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications—Include proceedings of conferences sponsored by NBS, NBS annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

Applied Mathematics Series—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a worldwide program coordinated by NBS under the authority of the National Standard Data Act (Public Law 90-396). NOTE: The Journal of Physical and Chemical Reference Data (JPCRD) is published quarterly for NBS by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements are available from ACS, 1155 Sixteenth St., NW, Washington, DC 20056.

Building Science Series—Disseminates technical information developed at the Bureau on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NBS under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The standards establish nationally recognized requirements for products, and provide all concerned interests with a basis for common understanding of the characteristics of the products. NBS administers this program as a supplement to the activities of the private sector standardizing organizations.

Consumer Information Series—Practical information, based on NBS research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.

Order the **above** *NBS publications from: Superintendent of Documents, Government Printing Office, Washington, DC 20402.*

Order the **following** *NBS publications—FIPS and NBSIR's—from the National Technical Information Service, Springfield, VA 22161.*

Federal Information Processing Standards Publications (FIPS PUB)—Publications in this series collectively constitute the Federal Information Processing Standards Register. The Register serves as the official source of information in the Federal Government regarding standards issued by NBS pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

NBS Interagency Reports (NBSIR)—A special series of interim or final reports on work performed by NBS for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Service, Springfield, VA 22161, in paper copy or microfiche form.

U.S. Department of Commerce
National Bureau of Standards
Gaithersburg, MD 20899

Guide to the Selection and Use of Fourth Generation Languages • Martha Mulford Gray • NBS SP 500-143